ANISKETCH: ALTERNATIVE AESTHETICS FOR COMPUTER ANIMATION TOOLS

BY

IAN JUN YEEN LOH

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the requirements for the degree
of Master of Design Innovation

School of Design
Victoria University of Wellington
2017
AniSketch
Alternative Aesthetics for Computer Animation Tools
Ian Loh
'Do everything by hand, even when using the computer.'

1 Hayao Miyazaki, cited in Solomon, 2006
Acknowledgements

Firstly to my supervisor Walter Langelaar, I am incredibly grateful for your guidance. Your patience know no bounds and your thoughts have undoubtedly shaped this thesis and myself for the better.

To the lecturers of the Computer Graphics Programme, John Lewis, Rhazes Spell, & Taehyun Rhee for showing the beauty and imagery of mathematics. And to my fellow CG peers for continuously raising the bar.

To the unwavering dedication of the staff at the Faculty of Architecture and Design.

Last but not least, to my partner Shanshan (who somehow put up with this whole business the entire time), my brother Bryan and Dad - I finished! Thank you for everything.
Abstract

The aesthetics of computer animation inherently draw from old modes of traditional animation production. AniSketch is an animation system that uses visual composition and sketching to produce animations, built in reaction to these established aesthetics of computer animation. Provocation through alternative tools is needed in order to incite the creation new aesthetics, tools and processes. The relationship between animation and technology is often marked by conflict that ultimately leads to the catalyst creative change. As digital media and digital tools become increasingly part of the lingua franca of the animation process, it is our empathetic understanding for the computer, its affordances and its relationship to the artist that will allow us to not only fully utilise, but also help us to continually provoke the status quo of the medium.

Keywords

Computer Graphics, Computer Animation, Software Studies, Computational Aesthetics, Sketch-Based Interfaces
Table of Contents

1. Introduction 8
   1.1 Value of Research and Research Objectives 9
   1.2 Research Methodology 10
   1.3 Definition of Key Terms 12

2. Literature Review 13
   2.1 The Computer Animated Image 13
   2.2 The Aesthetics of the Animated Movement 16
   2.3 Sketch Based Interfaces for Computer Animation 20
   2.4 The Aesthetics of Sketching in Animation 22

3. Design Precedents and Relevant Research 26

4. An Overview of AniSketch 30
   4.1 Movement Through Action 30
   4.2 Parametric Posing 30
   4.3 Visual Composition 31
   4.5 An Introductory Example - A Bouncing Ball 31

5. AniSketch Design Process 35
   5.1 Initial Proof of Concept - Maya Plugin 35

   Figure 5.1.A. A Screenshot Of The Maya Plugin Representing And Blending Keypoints

   Figure 5.1.B. (Top) A 1D Gaussian Curve With Its Height Values At Different Points. (Bottom) 2 Overlapping Curves Are Added Together

   Figure 5.1.C. Illustration Of Key Point Size And Overlap As The Distance Increases

   Figure 5.1.D. Plugin Window Showing Unexpected Deformation During Stroke Editing

   5.2 Standalone Software Prototype 40

   Figure 5.2.A. The Initial Processing2 Prototype Keylines And Keyshapes

   Figure 5.2.B. Illustration Of A Keyline. The Line Above The Two Drawings Illustrate The Different Interpolation Behavior
ANISKETCH: ALTERNATIVE AESTHETICS FOR COMPUTER ANIMATION TOOLS

Figure 5.2.C. (Left) A Practice Sheet From Calligraphy Class. (Right) A Drawing Of A Stroke From The Processing Prototype

Figure 5.2.D. Stylisation Of A Keyline And Keyshape In The Prototype

5.3 Final Design

Figure 5.3.A. Anisketch In Use

6. Animation Examples

Figure 6.A. Anisketch Graphical Notation Legend

6.1 Yet Another Bouncing Ball

Figure 6.1.A. Anisketch Notation For A Bouncing Ball

Figure 6.1.B. Assembled Animation For A Bouncing Ball In Anisketch

6.2 Walking An Elephant

Figure 6.2.A. Anisketch Notation For A Walking Elephant

Figure 6.2.B. Assembled Animation For A Walking Elephant In Anisketch

6.3 All Seeing Eye

Figure 6.3.A. Anisketch Notation For A Moving Eye

Figure 6.3.B. Assembled Animation For A Moving Eye In Anisketch

7. Discussion

7.1 The Aesthetics of AniSketch

7.2 Potential Applications

7.3 Limitations and Future Work

8. Conclusion

9. References
1. Introduction

What are the aesthetics of computer animation? And what are the benefits to building computer animation tools with alternative aesthetics?

The aesthetics of animation is often defined through its modes of production and the development of new technology to supplement it. Beginning with simple optical amusements such as the Zoetrope, the soundless, monochromatic nature of early film cells were successfully adopted by comic artists such as Winsor McCay, paving the format for early cartoons. The development of multiplane cameras and live-tracing techniques for 1937s Snow White were spurred by Disney’s desire to recreate the movement of living things and ushered in a new aesthetic rooted in realism. The 12 basic principles of animation, developed during this time is still widely regarded by animators today. The advent of computer graphics (CG) continues this progression by bringing animation into the digital realm. Supported by new techniques such as raytracing, texture mapping and 3D rendering, animators could now create entire worlds within the computer, freed from the physical and practical limitations. With the digital pixel’s power and flexibility, it leaves no surprise that animation has adopted the computer as its de facto tool, which has arguably also led to animation becoming a critical element of the media consciousness today.
However, as use of computer animation continues to grow, it becomes increasingly important and useful to understand how the unique qualities of the computer—its aesthetics—affect animation and its artists. There is already a varied body of discussion in this area, from digital hyperrealism to commentary of the perception of animation in film, to Deneroff’s Law, which suggests that more powerful and complex tools beget more complex animation. Within the same thread of these discussions, this thesis proposes the following:

The current user interfaces and techniques used by computer animation tools inherently impose aesthetic limitations on the artist. This is due to animation systems being designed for production orientated applications. In order to provoke new animation aesthetics and new applications of the animation medium, new types of animation systems need to be built.

In this paper, I examine the aesthetics of contemporary computer animation tools and its effect on the animation process. A new sketch-based animation system, AniSketch, is subsequently presented and discussed as a response to this analysis.

1.1 Value of Research and Research Objectives

The objectives for my research are as follows:

1. Establish a definition for the aesthetics of computer animation.
2. Examine the aesthetics of computer animation, and understand its essential qualities.
3. Develop an alternative animation tool in response to these aesthetics.
4. Discuss the viability and context of the developed animation tool.

The value of the research is twofold. First, is the development and discussion of a novel animation system. Second, is by discussion of the tool in the context of animation culture. In placing AniSketch within the discourse of established animation production practises, I hope to establish a theoretical framework that justifies the need for alternative animation tools beyond simple novelty.
1.2 Research Methodology

Research Through Design
During the initial stages of research, I enrolled in a set of Chinese calligraphy lessons to better understand the aesthetic qualities of brushstrokes in relation to movement. The skills and experience gained from the lessons was an important initial precedent during the design stage and helped me to establish a user interface aesthetic for later versions of AniSketch.

Prototypes
The initial proof of concept that led to this research was developed as a postgraduate project as a Python plugin in Autodesk Maya. Using that proof of concept as a base, a second prototype was built as a standalone application in Processing2. The second prototype was used to allow a broader exploration of visual aesthetics, as well as establishing what techniques needed to be developed in response to creating a full functioning animation system. The third and final version of the AniSketch design represented a complete rebuild of the code base that addresses several critical bugs, animation limitations, user accessibility issues and graphics optimisations.

Scenarios through Animation Examples
No formal user testing was performed with AniSketch due to scope and time constraints. However, to better understand the capabilities and limitations of the animation system, a series of descriptive animation example scenarios are provided. Some of the examples are based on animation outputs that might reflect the typical animation exercises animation may undertake, such as a bouncing ball and a walking cycle. It is the hope that these examples will help both users of the tool and readers of this paper to better grasp AniSketch's functionality and capability. Additionally, the example animations provide a basic framework for future user testing.

Evidence-Based Design
Some of the animation techniques implemented in AniSketch have already been explored in some form by other novel animation tools. The majority of these tools reside
within computer graphics papers, in the ACM Digital Library. These papers often discuss the implementation, justification and testing of these tools and have been incredibly helpful in both the development of AniSketch and the formation of my critical discussion.

**Literature Review**

An overview of relevant texts regarding new media theory, animation practice, and sketch-based interfaces is presented to help establish context for later arguments.

**The Three Paradigms of Human–Computer Interaction (HCI)**

*The Three Paradigms of Human-Computer Interaction* by Harrison, Tatar & Sengers (2007) have also helped to inform my research methodology. The paper formally introduces a 'third paradigm' to HCI as relating to 'meaning and meaning-construction' (p. 7). This in contrast to the first two paradigms, which namely concern aspects of ergonomics and information architecture. The authors argue that the use of objective measures, such task completion time cannot be used as a universal indicator for deciding how 'good' a design is. As the authors put: 'Instead, we must ask questions about what it means for a system to be 'good' in a particular context...' (p. 8). The value of this context is apparent in the many of the computer graphics papers reviewed here. In an evaluation of *Spatial Keyframing for Performance-driven Animation* by Igarashi, Moscovitch & Hughes (2005), the authors describe that while despite limitations of the tool, the users 'commented that the system was fun to use and the experience was very novel' (p. 8). *The third paradigm of HCI*—similar to design ethnography practice—helps to establish value of such comments within the space of computing.
1.3 Definition of Key Terms

The Aesthetics of Computer Animation

In the field of computing, the word aesthetics has been used to describe different ideas. For this reason, it is worth briefly clarifying the definition of what is meant by aesthetics in the context of computer animation, and within this paper.

In the review On the Origins of the Term Computational Aesthetics, Greenfield (2005) traces early use of the word aesthetics within computing to describe the notion of an aesthetic measure for works of art - that is, the development of mathematical models and metrics to understand and classify the qualitative values that we place on art, design and other forms of imagery. More recently, the term computational aesthetics has been tied to the idea applying computing techniques such as neural networks and genetic algorithms to the task of discerning and acting upon aesthetic qualities in imagery (Greenfield, 2005). These ideas are a common thread in computer graphics research, such as the case of editing a picture to match a mood, classifying 3d objects by shape or mapping the qualities of motion data from one model to another.

In Perspectives on Aesthetic Computing, Fishwick et al. (2003) defines the field of Aesthetic Computing by examining the converse, extending the idea of aesthetics within computing to include 'the theory and application of art to computing'. For example, Aesthetic Computing could involve representing data structures using cultural metaphors, or approaching user interaction design through lens of artistic concepts such as Gestalt. Moreover, the authors stress the idea that this relationship is symbiotic:

'Aesthetic computing implies that art is affecting - and reflecting - some aspect of computing... It may be that taken to its logical conclusion, art affects computing and that this, in turn, affects art, closing the loop: art creates art.' (Fishwick et al. 2003, p. 1)

While Greenfield's definition of aesthetics is arguably the more common one within computer graphics, the emphasis placed on the reflective relationship between art and computing by Fishwick et al. aligns more closely with the discussion in this paper.
Building upon the ideas of *Aesthetic Computing*, I define the *aesthetics of computer animation* within this paper as the following:

*The modes of interaction and practises that stem from the interdependent relationships between computer graphics, animation media, artist methodology.*

## 2. Literature Review

### 2.1 The Computer Animated Image

Unlike film, which can be seen as a record of realism at its essence (Manovich, 2001, p. 295), computer generated imagery (CGI) represents an ability to synthesize visual reality. These realities can be subjective, for example the reality of a cartoon is just a valid as the reality of the physical world. Despite this, much of the visual identity of CG is characterized by *photorealism*: the reality as it exists within the frame of the camera (p. 184). Others like Power (2009) also offers the term *naturalism*, referring to general idea of pictorial verisimilitude that 3D computer graphics aspires to (p. 109). For the purposes of this paper, these terms are used interchangeably. Manovich (2001) attributes the development of the photorealistic aesthetic to early sponsors of computer graphics research: the military and the movie industry. The requirements set by these groups ’led researchers to concentrate on the simulation of the particular phenomena of visual reality, such as landscapes and moving figures’ (p. 193). Moszkowicz (2002) points to another potential reason for a realist tendency; recreating naturalistic phenomena allowed early experimenters a way to get to grips with the new medium and provided an accessible benchmark to compare their work with (p. 299).

Today, the toolsets of computer animation packages still focus heavily on creating photorealistic imagery. This may be in part due to the overlapping uses of these tools in applications such as computer-aided design (CAD), medical imagery and simulation. Furthermore, the generation of special effects, which by nature requires seamless integration with live-action footage, is also a large driver of realism within 3D animation (Power, 2009, p. 110). In the case of 3D animation, the aesthetic qualities of elements such as lighting and texture are largely determined by physically-based renderers,
unlike 2D animation where they may be distorted to better fit artistic intentions. As rendering a 3D scene is a time consuming process, artists will work with a low fidelity representation of their scene. However, this dissonance may lead to a difference in how the animation is perceived after it is rendered. Lasseter (2001) points out that different stylistic approaches are required when working with realistic looking 3D characters. He contrasts the technique of ‘holding frames’ in 3D animation:

“In 3D computer animation, as soon as you go into a held pose, the action dies immediately . . . It must be the combination of the dimensional, realistic look and the smooth motion (usually on ‘ones’) that makes a hold cause the motion to die. The eye picks it up immediately; it begins to look like robotic motion.” (Lasseter, 2001, p. 46)

To address this, Lasseter suggests that animators should make the different parts of the character's body come to a stop at different times. Contemporary animators such as Jason Figliozzi, who has worked on Paperman (2013) and Big Hero 6 (2014) echo a similar importance of subtle motion in keeping the illusion of realism: ‘the difference between whether something seems real or not can often hinge on the smallest movements’ (Figliozzi, as cited in Collins, 2015). In other words, an animation that looks good in preview may be lacking when fully rendered.

In Character Animation and the Embodied Mind-Brain, Power (2008) describes a study of fMRI scans of participants while watching Richard Linklater's film, Waking Life (2001). Participants watched two versions of the film: the original live-action edit of the film, and the subsequent rotoscoped version that was released. Participants who viewed the live-action video showed activation in the areas of the brain involved in the reading the intentions of others, while viewing the stylised rotoscoped animation activated areas associated with emotional reward (p. 40). While Power notes that the interpretation of the results are not definitive, it implies that visual style plays an important role in the perception of movement, and can may explain why realistic visuals are less forgiving when paired with certain styles of animation (Power, 2009, p. 115).
Notwithstanding, animators are not necessarily dictated by this photorealistic-technological framework. As evident from the stop-motion inspired *The Lego Movie* (Lord & Miller, 2014), distorted motion capture from *going to the store* (Lewandowski, 2012) or ‘psycho-realism’ style of *Ryan* (Landreth, 2005), photorealism can be appropriated to create unique animation aesthetics. In addition, the field of non-photorealistic rendering (NPR) is rapidly maturing as researchers look to artistic processes outside the realm of simulation, providing new avenues of visual treatment for 3D animation (Gooch et al. 2010). Conversely, a good animation is not necessarily dependant on its visual fidelity; *Toy Story’s* (Lasseter, 1995) characters are endearing to us for their well written dialogue and crafted movements despite their relatively low fidelity appearance by today’s standards.

However, there is a case to be made about about the ostensibly deep but narrow range of computer animation’s image making facilities and its effect on the animation process. Computer animation tools approach animation from an image orientated viewpoint, where movement and image are inseparable. While this may seem an intuitive association, Moszkowicz (2002) points out that early explorations of animated media tended ‘celebrate disjunction over fusion, maintaining a distinction between image and object, animation and world’ (p. 296). Indeed, long before Disney popularised the classical look of animation with the likes of *Snow White* (1937), works by pioneers such as Oskar Fischinger recognised the expressive potential of motion without image through abstract shapes and forms. Torre (2014) argues that the distinction between image and motion is not only an essential quality of the animation process, but also that ‘the animator must at least have an awareness of the pure motion, distinct from the form that they are engaged with, in order to create a convincing movement of that form’ (p. 52). He continues that the unique digital representations of motion, such as motion performance capture allow movement to combined to any image, creating new meanings through recontextualisation. Looking to the fields of motion graphics, data visualisation and generative art, we are offered glimpses of a new aesthetic built upon these tenets, while *The Dot and the Line* (Jones, 1965) show us that animations do not need classical Disneyan characters to be affective.
2.2 The Aesthetics of the Animated Movement

Callahan (1988) notes that the modes of production for early animation practitioners was a long and tedious process. Each frame was drawn by hand and several months of work would only result in a few minutes of film (p. 224). To remain competitive, studios standardised the animation process, creating specialised roles and a division of labour (p. 225). Of these new roles, most critically for our discussion is the position of the inbetweener. As the name suggests, the inbetweener was responsible for taking a few key drawings provided by the animator, and fill in the frames in between those drawings required to create a smooth animation, freeing the animator to focus on designing characters and their movements (Tai, 2013, p.116).

Disney in particularly was known for adoption of these techniques for large scale production, and Tai (2013) asserts that the segmentation of the animator into smaller roles led to a standardisation of animation aesthetics. This is apparent when contrasting the studio’s cartoons production prior and leading up to the beginning Disney’s Golden Age, as the studio rapidly expanded to produce Snow White (1937). While both 1932’s The Mad Dog and 1936’s Mickey’s Elephant show the character Pluto in various states of distress, the movements in the latter are noticeably smoother and more coherent. In contrast, the former appears more spontaneous and varied, featuring sudden changes in pace and wild body distortions. Arguably, the decline of these aesthetics are as much attributed to Disney’s desire for more naturalism as its incompatibility with the studio’s new production methodology. The splitting of the animator’s role also paved the way for computers to gradually automate and supplant many aspects of the animation production. Notably, researchers were quick to recognise the labour intensive and time-consuming nature of inbetweening, introducing automatic inbetweening techniques inspired by the same concepts (Tai, 2013, p. 118).

Automatic inbetweening or keyframe animation is the by far the most common technique used in computer animation. Keyframe animation involves setting the states of an object—its position or colour for example—at specific frames. These keyframes are then automatically blended (known as interpolation) from one to the other by the computer to produce a smooth transition. Alternative computer animation techniques
such as motion capture and parametric animation, as well as other forms digital media also utilise keyframes for storing changes over time. Despite this, keyframe animation retains the aesthetic qualities that are symptomatic of its origins—in particular the ideas of smoothness and segmentation—that are further exacerbated by its digital representation interpolation and correspondence.

Unlike a traditional inbetweener, keyframe animation has no context of the artist's intentions with regards to the speed of the interpolation between keyframes. Because of this, animations often start out with a default linear or cubic interpolation applied to all keys, creating an overly smooth and predictable motion. The result is unconvincing, exhibiting a sense of weightlessness that Tai (2013) describes as ‘awkward agility’ (p. 112). Lasseter (1987) describes the speed on of an action - otherwise known as timing - to be a critical element to animation, 'the speed of an action defines how well the idea behind the action will read to an audience. It reflects the weight and size of an object, and can even carry emotional meaning' (p. 37). The timing of a keyframe’s interpolation is typically represented as a spline and can be manipulated by the artist to create a more expressive output. However as Lasseter (1987) notes, the smooth nature of a spline's shape can interfere with the animation by 'overshooting', where a large change causes the spline to curve in the opposite direction a little before returning back. To correct this, more keyframes are needed to ‘force’ the spline into the right shape (p. 41). This additional effort appears to run counter to the idea of automated inbetweening and as Shannon (1997) points out, the spline interface is an abstract, mathematical and ultimately dislocated representation of the final motion, and as such, may be difficult to intuitively interact with (p. 435).

In *Principles of Traditional Animation Applied to 3D Computer Animation*, Lasseter (1987) describes 2 types of animation approaches: *straight ahead* and *pose-to-pose*. In straight ahead, frames are drawn sequentially with only a rough idea of what action takes place. The animator incorporates new ideas into scene as he draws, creating 'a fresh and slightly zany look'. According to Lasseter, 'straight ahead action is used for wild, scrambling actions where spontaneity is important’ (p. 40). In contrast, pose-to-pose involves planning of the character's movement in advance, making drawings of the important poses, and then drawing the inbetweens. Lasseter makes the
comparison between pose-to-pose and keyframe animation in the sense of breaking the motion into of set of poses. He also points out that keyframe animation needs to be animated in a layer by layer fashion, starting from the ‘trunk’ and working outwards, noting that planning very important or the animation will quickly lose its clarity (p. 40). As seen with the traditional inbetweening process, this structure and need for planning makes it difficult to produce the more freeform movements that characterise straight ahead animation.

Catmull (1978) establishes that the main difficulty of computer driven interpolation as finding the correct correspondence between keyframes. This is especially the case for hand-drawn characters, where information is lost if parts of the body are covered or if characters rotate. Catmull outlines 3 main ways to approach the issue - break characters into smaller pieces, restrict animation range to minimise issues or manually tweak frames as needed (p. 350). While the keyframe represents a segmentation of motion that led to smoother but more constrained movement, the segmentation of characters into components in order to utilise keyframe interpolation has had led to easier production at the cost of dimensionality of characters. Tarantini describes how the shift to digital pipelines prompted a change in the treatment of characters:

“The new production pipeline made use of extensive visual ‘asset libraries’ that were able to facilitate and accelerate production while maintaining continuity in style and design . . . Characters were designed, coloured and separated into elements (assets): heads, torsos, arms, legs, hands, mouths, eyes and any other part of the character that may need to move.” (Tarantini, 2011, p. 260)

George Elliot, founder of Toronto-based Elliot Animation, refers to the popular vector based animation system Flash as a ‘mixed blessing’ (Tarantini, 2012, 10:14). Since characters were animated by manipulating a set of pre-made components, junior animators could produce relatively high quality animation quickly. However, the limited set of components restricted the range of movement to certain dimensions. In computer 2D animation, movement is often constrained to the lateral plane (side to side), making it difficult for artists to express in and out movement through the frame, which was
more commonly seen in hand-drawn animation. In 3D animation, characters are usually driven by a skeleton rig, a set of digital 'bones' that are placed inside the character that deform the 3D mesh. While 3D characters can be depicted from any angle and thus do not have the same issues of dimensionality, enforcing the use of a digital joints makes 3D animation conceptually incompatible with -or at least difficult to work using- animation styles that are characterised by the lack of joints, such as rubber-hose or smearing.

Along with keyframe animation, motion capture is another popular technique for creating digital movement. Sturman (1994) defines motion capture as 'the recording of human body movement (or other movement) for immediate or delayed analysis and playback' (p. 1). The idea is not new in animation; Disney was well known to use light tracing techniques to 'capture' the motion of human actors in video footage for the characters in Snow White (1937). However, Moszkowicz (2002) notes that the technique was not received well at the time, with Disney 'perceived to have hijacked a potentially expressive medium and subsequently to have set limits on its horizons' (p. 310). Ken Cope quoted in Cameron et al. (1997) raises similar concerns in regards to motion capture:

'Motion capture is more closely related to the cinematographer’s medium than to that of the animator. Typically lacking even the slightest benefit of the animator’s unique expertise, a director must coax a performance (the one he’ll know he wants when he sees it) from a capture-rigged movement specialist. The motion is recorded, not created, by a technical director ... Too many aspiring animators find their job options limited to the role of captured data processors.' (Ken Cope, as cited in Cameron et al., 1997, p.2)

Cope argues that because motion capture is more like acting, that it devalues the worth the animator. However, the analogy can also be seen as complementary to animation process as acting is often employed by animators to better understand the motion they are about to create. In a production video of Persepolis (2007) Vincent Paronnaud, remarks of fellow co-director Marjane Satrapi: 'Animators need some clues from the
filmed screenplay (animatic), so they go and see Marjane. Marjane will mime the grandma, the dog, the table, for example’ (Paronnaud, as cited in Film Education, n.d.)

Instead of having to refer to notes or a video recording, motion capture provides a way for animators to directly utilise the qualities of natural motion in their animations. This can be seen in tools such as Dragimation by Walther-Franks et al. (2012) that use the animation timing in real world movement to drive the timing of animation, addressing some of the difficulties in working with the abstraction of keyframes (p. 101). As motion capture is simply a recording, a complex action is just as easy to record as any other action. Additionally, it is able to retain subtle qualities about the motion that may be noticed by the animator. In keeping with Lasseter’s comparison to traditional animation methods, this affordance of fidelity and freedom can be seen as compatible with the idea of straight ahead style of animation.

However, while motion capture techniques offer a counter-balance to the aesthetics of keyframe animation, Thalmann (2008) points to several practical issues regarding motion capture. Aside from software and hardware costs required, motion capture is limited by the physicality of space and its data difficult manipulate for other uses. Additionally, Thalmann argues that the processing of motion capture data to build systems of motion, rather than using the raw data itself, is essential to create effective digital movement; motion capture is only the first step, or at least one step in many.

2.3 Sketch Based Interfaces for Computer Animation

AniSketch relies on the action of drawing to create computer animations. The use of pen-based input in computing precedes even the invention of the now ubiquitous mouse by several years (Jorge & Samavati, 2011, p. 2). Sutherland’s (1963) pioneering work on Sketchpad utilised a light-pen in order to create engineering drawings without the need to type in commands, which was the norm at the time. Sutherland’s argument was that communication through a series commands inhibited the interaction process: ‘we have been writing letters to rather than conferring with our computers’ (p. i). In Sketchpad, drawings made on the screen would be beautified and displayed as clean lines at interactive rates. In combination with operations such as copying, constraints and alignment of drawings, Sketchpad enabled broad applications from visualising
mechanical linkages to designing repetitive patterns (p. 7). By removing the need for explicit commands, Sketchpad offered a new approach to human-computer interaction, one where intuitive modes of input were complemented with procedural techniques and computer algorithms to infer them. Such systems not only have the potential to increase productivity, but also foster exploration through new forms of digital manipulation.

The foundations laid by Sketchpad have seen a revival since the 1990s, in the field of sketch-based interfaces (SBI) owing to faster computing and availability of pen-based technologies. SBI focuses on the use of 'sketching' - drawing action using a pen, mouse, touch or otherwise - to enable fluent interaction between computers, usually for graphical applications. Consider the example of drawing a simple rectangle rotated at 45 degrees. In a typical vector application, one might imagine a series of actions like:

1. Click on the 'create rectangle' button
2. Click and drag to create the rectangle
3. Resize the rectangle to desired dimensions
4. Rotate the rectangle 45 degrees

Using SBI, one could simply provide a sketch of the rectangle, and the computer would infer the appropriate position, rotation and dimensions of the drawing in a single gesture.

Today, digital artists often incorporate a stylus in their arsenal of tools. Styluses allow greater fidelity in the form of better precision and added expressivity in the form of pressure and pen orientation. Despite this, support for stylus input in professional grade software is limited to only a few cases. While artists may be able to digitally paint or sculpt models, their canvases remain surrounded by zoo of buttons, sliders and dialog boxes, otherwise known as the WIMP (windows, icons, menus, pointers) interface. Neilson (1993) argues that WIMP interfaces are categorically no different from early text-based command interfaces in that they are 'based on the concept of an explicit dialogue between the user and the computer during which the user commanded the computer to do something' (p. 84). In other words, bar a few cases, pen input is handled akin to a glorified mouse pointer—the affordances of sketching have yet to be fully
utilised in modern digital tools. WIMP interfaces may be desirable for animation production environments where total control is paramount. However, Power (2009) acknowledges such control may pose creative obstacles to artists themselves:

"It is a lot to ask of animators (or anyone else) that they should excel in visual and motion design, character design and storytelling, and then be capable programmers as well, so the implementation of intuitive interfaces for algorithmic design processes is critical. Many modern interfaces are still encumbered with numerical or other non-intuitive forms of input, often into a bewildering host of variables with indeterminate functionality . . . What might be called a magical-realist approach to interface design would be optimal in this context, with functionality based on real-world analogues where helpful, but with ‘magical’ digital capability besides." (Power, 2009, p. 125)

Sketch-based interfaces offer an approach to the 'magic-realist' design that Power refers to. While computer can be seen as the de facto tools for the typical animation production, drawing remains an important skillset behind the scenes. SBI can serve to bridge the gap between computer animation with exploratory processes and other forms of animation media. In the following section, I will discuss some of the ways that gap is addressed and perceived by examining the relationship between animation and 'sketching' (as well as its many derivatives).

2.4 The Aesthetics of Sketching in Animation

Above animation, above movement, above style, is the ability to draw - in particular with traditional or classical animation. Art Babbit, known for his work on Disney classics Pinocchio (1940) and Fantasia (1940), among others, puts it bluntly: 'If you can't draw - forget it. You’re an actor without arms and legs' (Babbit, as cited in Williams, 2002, p. 25). Babbit implies that mastery of the line is the central pillar of expression in animation. Shannon (1997) compares the master animator’s ability to fluidly lay down lines from his or her imagination, to a performing art, and source of an animation’s
identity (p. 434). Through Shannon’s comparison, drawing can be seen as a form of motion capture where the animator’s hand becomes the actor.

This characteristic mark of the artist is an important element to the storytelling process. Persepolis (2007) director Marjane Satrapi remarks that the ‘vibrations of the hand make the drawings come to life’, arguing that the natural variations of the line are reflective of the imperfect characters in the film (Satrapi, as cited in Film Education, n.d.). Aardman Animation’s supervising director Richard Goleszowski reaches similar conclusions: ’I think the fact you know it’s a hunk of plasticine and occasionally you can still see the fingerprints—some of the process is revealed and that actually helps you tune into the character’ (as cited in Strike, 2007). Research by Freedberg & Gallese (2007) propose that these visible gestures are able to induce empathetic responses in observers by activating areas in the brain that simulate that action that created viewed gesture (p. 202). While Power (2009) describes computer animation as typically lacking such mark making qualities, he suggests that gesture based interfaces (such as those in SBI) may ‘lend a reflexive element to animating’ and facilitate ‘feeling the movement behind the mark’ (p. 125).

Line art outside the realm of animation also exhibit the qualities of ‘movement behind the mark’ and are equally useful to include in this discussion. The art of calligraphy offers a wealth of visual inspiration - far from being simply utilitarian, the movements of the pen or brush are recorded in the shifting thickness and saturation of lines. Chinese and Japanese calligraphy in particular emphasize importance on the channelling the artist’s physical and mental state through the brush and onto paper. As Gunn (2001) describes:

‘The brush accurately mirrors your own mental and physical attitude. Rigid thinking produces rigid brush strokes . . . When mind and body are unified, the eyes are calm and clear, the muscles quick and responsive. This quality is transmitted to the brush, and becomes visible in the brush strokes. Slackness reflects a discontinuity between mind and body, and results in poor control of the brush . . . the brush becomes a veritable extension of the
self, completely and finely sensitive to every nuance of the body, which is the complete expression of the mind.' (p. 156-157)

The depiction of movement, space and emotion is well explored in calligraphy. Brushwork is often arranged in strong visual-spatial compositions aimed to suggest scenes through the thoughtful and sparse use of strokes. Such compositions are driven by what Kwo (1990) describes as the 'psychological consciousness of space constructed by the universal faculties of perception, such as seeing and feeling'. This calligraphically created space is an attempt to depict the essence or the poetic nature of the scene through 'the gesture and rhythmic indications of line' (ch. 6, p. 15). A few effective strokes may be enough to suggest a swimming fish, falling leaves, or the feeling of wind.

A similar visual language can be seen in animation production, albeit admittedly in a less abstract fashion. Storyboard artists work with directors and writers to help visualise the story through rough thumbnail sketches. Rough pencil tests, a collection of loose and imprecise lines, serve to help animators explore the tone of performance. Animation concepts such as 'line of action' and 'arcs' ensure effective character poses and movements by following smooth, pleasing, shaped curves (Williams, 2002, p. 90; Blair, 1994, p. 90). Complex characters may warrant the creation of control 'rigs' to aid animators, a series of interactive shapes, sliders, buttons that control different parts of the character. These control objects are often arranged in meaningful, and visually pleasing ways in relation to the character. Onion-skinning, a display technique where several frames of an animation are overlaid together, give a similar impression to a calligraphic brush, as the shape of the animated object traces a 'line' that changes thickness and opacity in response to the object's movement in space. The animation technique of the 'elongated inbetween' where characters are smeared in relation to the movement, also share 'brush' like qualities (Williams, 2002, p. 96).

The speed and fluidity afforded by the sketching process also plays an essential role in supporting creativity. In a series of case studies, Austin, Devin & Sullivan (2012) suggests that the conduciveness to innovation depends largely on whether original outcomes are beneficial, and if originality is inexpensive enough to produce (p. 1509).

As discussed in earlier sections, creating a computer animation using techniques such as
keyframing are ill suited for quick ideation, as it involves planning in advance and many adjustments thereafter to get the motion right. However, a series of rough thumbnails or pencil tests would be very fast to draft out and be able to communicate the idea just as well. Byron Howard, co-director on Zootopia (2016) describes the ability to quickly cycle through ideas as an important part of shaping the film’s identity:

‘All of us who board—who storyboard—on this movie have done literally thousands of drawings that have wound up in the garbage because these ideas come in and they go out of the films very, very quickly, and the idea with that—is to be wrong as quickly as possible so we really try to, you know, try out as many idea as we can and then when the right ideas really click in - then you’ll see from screening to screening they’ll start to build, the rest of the film will start to build from those key ideas.’ (Howard, as cited in Osma, 2016)

Kim, Bagla & Bernstein (2015) and Owen (2012) argue that failure and accidents are underpraised in their role in facilitating creative outcomes. Kim et al. warns that current creative tools are designed for ‘showcasing success or critiquing finished work, rather than providing safe spaces for experimentation’. For less experienced users working with these tools, the intimidation and fear of failure prevents them from engaging with opportunities for growth (p. 157). Activities such as sketching in contrast have ‘weak filters for success’, where mistakes and explorations tend to more readily accepted by the artists and others (p. 158-159). The looseness and propensity for mistakes in sketching also lend itself to what Owen (2012) describes as ‘accidental mark making’. Owen gives the example of how we are quick to interpret a child’s early uncoordinated scribbles as a ‘person’, or a ‘flower’ despite the drawing bearing no resemblance to the either (p. 322). Owen posits that these reduced drawings are ‘hyperreal’ images in that the lines represent more than is physically depicted, and our natural ability to perceive them allows for new stylistic explorations of our perceived reality (p. 325).
3. Design Precedents and Relevant Research

The action of putting pen to paper is a familiar experience for many and sketching is often still used as way to develop initial ideas and prototypes. Today’s digital tools allow users to realise their ideas with more visual and functional fidelity. However, the higher quality and extra effort needed to execute these ideas digitally impose unnecessary obstacles to testing and ideation process (Rettig, 1994, p. 22). Additionally, the skill required to use these tools also put them out of reach of casual users. By leveraging the natural language of drawing, sketch based interfaces can make digital tools more flexible and accessible for a variety of applications and users. The general drawback of implementing a sketch-based interface, is the loss of precision typically afforded by more traditional user interfaces. The ambiguous nature of sketching could also lead to users getting frustrated if errant inputs are handled improperly. Igarashi (2003) advises that these issues can be mitigated by giving user the right impression through a more informal visual aesthetic, and making tools fast enough to facilitate trial and error experiences (p. 47). For these reasons, SBI is typically adopted for creative exploration tasks rather than for production based applications.

In the modelling tool T.E.D.D.Y by Igarashi et al (1999), users can create 3D models by drawing simple shapes. Operations such as slicing and smoothing a surface are executed through contextual gestures on top the model. The authors note that the tool was easy enough for children to use, and the interface led the models to have a unique 'hand-crafted' aesthetic, which would be difficult to achieve in more professional modelling software. SILK by Landay & Myers (1995) is an interface design tool that recognises sketches of UI widgets and applies the appropriate behaviour to them. The "sketchy" look is retained, as not to distract users from the task of evaluating the functionality of the design, rather than the 'fit and finish'. By blending hand sketches with the functionality of more traditional interface design tools, users of SILK can very quickly prototype ideas by drawing their ideas as they would on paper.

Animation software packages like Autodesk’s Maya and 3ds max are specifically built for professionals in the industry. Even freely available animation tools such as Blender...
pander to production orientated applications. On the other end of the spectrum, there are surprisingly few tools in the public sphere that are aimed towards casual animation creation. This unbalance has led to extensive research in developing alternative animation tools that utilise sketch-based interfaces. These applications can be broadly divided into two categories - firstly, the manipulation of objects or character poses in preparation for animation and second, the generation of movement for the actual animation itself.

Digital character manipulation is often done through puppet-like control rigs and sliders. These controls are not necessarily intuitive and animators will often sketch the poses on paper in advance. In this thread, Davis et al. (2003) and Matthews and Vogts (2011) have proposed tools that use a stick figure style notation to drive character poses. Guay, Cani & Ronfard (2013) further simplifies this process, using the animation concept of 'line of action' to pose characters using just one or two stylised curves. The authors note that higher-level abstraction allows the tool to be used in earlier stages the creative process, where the focus is on the overall pose of the character. Sketch interfaces have also been developed for posing character faces such as proposed by Chang & Jenkins (2006) and Miranda et al. (2011), using drawn strokes to indicate how a face should be deformed. Notably, while the tool from Chang and Jenkins use strokes drawn on top of the model, Miranda et al utilise a separate drawing region as the control interface. The separation from the models allows the strokes to be saved and reused for other models. In traditional animation systems, movement is inherently linked with an object. The AniSketch control interface similarly decouples the motion from objects, storing movement in the form of strokes that can be reused with other keys.

The second application of SBI is in the generation of movement. Motion recording tools proposed by Moscovich & Hughes (2001), and K-Sketch by Davis, Colwell & Landay (2008) leverage the concept of motion by example, wherein the movement of objects are recorded and their actions replayed as an animation. As opposed to setting keyframes, user’s hand movements are utilised, allowing them can act out the motion. In their evaluation, Moscovich & Hughes (2001) report that the tool may be able to better
preserve an animator’s personal timing. The tool also provided a different aesthetic that was apart from other animation tools such as Adobe Flash.

Performance-based animation introduced by Terra and Metoyer (2004) and extended by Walther-Franks et al. (2012) also allows users to act out movement. The notable difference is that poses are still set with keyframes beforehand, and the timing of the animation is controlled by drawing a line with the desired speed. Performance based animation can be seen as a subset of motion capture, and allows animators to combine the natural expressiveness of his or her body with the precision of keyframed poses, allowing for high degree of artistic control. However, the quality of the animation is also relies on the user’s acting capability, and ergonomic limitations of the body.

Animation tools by Igarashi, Kadobayashi, Mase & Tanaka (1998), Thorne, Burke & Panne (2004) and Jang, Jeon, Sohn, Lim & Choy (2013) adopt a more ‘directorial’ style approach to animation, where a character moves along a path that is drawn onto the scene. Further drawings can be made to refine the movement of the character. Motion Doodles by Thorne et al. (2004) uses a set of notations that indicate different types of movement. These notations can be written in a cursive fashion to create a smooth series of movements. Importantly, the drawings also represent a meaningful visual record of the animation, which is absent in acting based systems. On the other hand, while animation using drawn gestures are fast and easy to create, they are often more restrictive in the range of the animations possible. Each new type of movement requires a unique gesture, and a predefined library of animation which naturally can only be applied to certain types of characters. Additionally, drawing on top of a scene presents additional challenges such as occlusion and distortion caused from perspective.

These techniques are not wholly exclusive. DRACO by Kazi, Chevalier, Grossman, Zhao & Fitzmaurice (2014) uses a combination of drawn gestures for general motion, and direct manipulation for finer adjustments. AniSketch most closely resembles performance-based techniques in that drawn strokes are also used to adjust the timing of predefined keys. However, like high-level gesture based tools, AniSketch displays drawn strokes on screen in order to help the animators develop a ‘visual vocabulary’ for
creating animations. The visualisation of the strokes are also notably different, employing an ink brush aesthetic that is reflective of Igarashi’s idea of informal presentation (2003, p. 43).

AniSketch uses a spatial keyframing system as its basis for defining poses. Spatial keyframes differ from traditional keyframes in its dimensionality. Traditional keyframes are placed along a timeline, which is essentially a 1 dimensional space. Thus, each keyframe is able to only interpolate between its neighbours. By contrast, spatial keyframing places keyframes in a 2D or 3D area, allowing interpolation between many different keyframes at once. The added dimensionality allows for expressive and emergent animations to be created using only a few poses. Animations are created by moving a cursor through the keyframe space, and the resulting animation is created by blending the poses based on distance between the cursor and its respective keyframe. Spatial keyframing tools presented by Igarashi, Mosovich & Hughes (2005) and Baxter, Barla & Anjyo (2009) employ radial basis functions or mesh based interpolation techniques to smoothly blend between keyframes. The blending parameters of the keyframes in these systems are automatically adjusted to allow for easy and intuitive exploration of poses. AniSketch uses a naïve approach to interpolation - by simply adding the poses weighted by the distance of the cursor to the keyframe. Instead, users are able to edit the keyframe’s area of influence and join multiple keyframes to create more complex keyframe 'shapes'. AniSketch relies on the user’s ability to determine how the interpolation behaves. While this adds some difficulty in the initial setup of the tool, it enables more customised and flexible control interfaces. To support this, AniSketch visualises the keyframes in a style of a customisable 'heat map'. The projection of keyframes onto 2D surface allows sketching to be used as an input for spatial keyframes. Furthermore, the ability to abstract the control interface serves to some mitigate the limitations of performance-based or acting-out techniques.
4. An Overview of AniSketch

AniSketch is a tool built to explore alternative techniques and interfaces for creating computer animation. Conventional animation tools are typically designed with high quality production in mind, affording a wide variety of precise controls and functionality to artists. While this high degree of control is often desired, the time, effort and skill needed to use these tools makes animation inaccessible to casual users and for applications outside of production animation. Furthermore, the focus on production and the need for interoperability leads these tools sharing a relatively standardized set of user interfaces and computing techniques that can impose aesthetic restrictions to users. In contrast, AniSketch focuses on animation for creative exploration, allowing users to quickly create movement through the combination of simple but flexible components coupled with a more pared down and intuitive interface. AniSketch breaks away from the conventions of typical animation tools in 3 key ways, through the use of movement through action, parametric posing and visual composition.

4.1 Movement Through Action

Creating animations in AniSketch centers around the sheet—an area in the software where the user can create movement by drawing using the mouse or a digital stylus. The sheet acts as a control interface as the user draws on the sheet. The position of the drawing cursor is used as an input to drive an object's movement in real time, allowing the user to utilize their natural sense of natural timing to 'act out' the objects movement, and react accordingly the result.

4.2 Parametric Posing

The other key component of the animation system are keypoints. Keypoints reside within the sheet and hold information about an object's attributes, such as position or rotation. These attributes are smoothly blended together based on the keypoint's distance to the drawing cursor—the closer to the cursor, the stronger that attribute's influence. The user is free to determine the placement and size of keypoints on the sheet, and can connect multiple keypoints together to form lines, triangles and other
shapes. Object attributes are stored as relative changes (turn clockwise by 90 degrees) rather than absolute values (the rotation is 90 degrees). These qualities allow the users to combine and remix object attributes in many ways to create wide variety of poses with only a small set of keypoints.

4.3 Visual Composition

Animations in AniSketch effectively have 2 representations, the resulting animation, and the underlying arrangement of drawn strokes and keypoints. The latter representation is given a strong visual treatment to provide the user with a unique perspective on their work. Drawn strokes are rendered with an ink like texture to convey the style of motion with the thickness and opacity based on speed. Keypoints show their area of influence by shading the sheet in different colours.

4.5 An Introductory Example - A Bouncing Ball

To illustrate an application of the AniSketch system, consider the basic example of a bouncing ball. For clarity, we will be using a simplified graphical notation to show the different components used in this example animation.

A bouncing ball exhibits an oscillatory motion that decays with each subsequent bounce. To begin, we will create the basis of this motion by drawing a wavy stroke that flattens out over time (Fig.5.5.A). Each 'peak' on this stroke will represent a separate bounce. AniSketch records this stroke as a line that can be replayed later:

![Figure 4.5.A. Drawing a Wavy Line](image-url)
Next is the creation and placement of keypoints. The starting scene comprises of a single ball at a rest position. In AniSketch, this rest position is referred to as the default pose. Keypoints store changes relative to the default pose and all animation is calculated as the result of the default pose plus any influences from other keypoints. How much influence a keypoint’s pose contributes at any point depends on the distance between to that keypoint's center. The influence is strongest in the middle of the keypoint, decaying outwards towards zero. In graphical notation, a dotted border is used to indicate a keypoint's area of influence. The figure below shows an illustration of the keypoint and its influence.

![Diagram of keypoint and influence](image)

**FIGURE 4.5.B. GRAPHICAL NOTATION OF A KEYPOINT AND ITS INFLUENCES**

To create the bouncing ball animation, a keypoint is positioned on top of every 'peak' of the stroke. Each keypoint is set with a pose that tells the ball to move upwards from its default pose, moving less for keypoints with shallower peaks and more for higher peaks (Fig 4.5.C).
The animation is now complete. The animation is created by retracing the drawn stroke, calculating the influences of every keypoint as line is traversed. In the figure below (Fig.4.5.D), we can see exactly how the motion is created. Starting from point 1, the ball is floating in the air since Pose 1 is exerting its maximum influence. In point 2 and 3, Pose 1’s influence decreases until it no longer has any influence, leaving the ball firmly on the ground at its default pose. Point 4 and 5 reenters another pose - Pose 2 - which moves the ball upwards again but a little less this time around. The process continues,
weaving in and out of keypoints until the ball finally comes to a complete rest at point 15 and 16.

FIGURE 4.5.D. VISUALISED INTERPOLATION OF BOUNCING BALL ANIMATION
5. AniSketch Design Process

5.1 Initial Proof of Concept - Maya Plugin

The original concept that led to AniSketch was inspired by the practice of animators 'acting out' the movements of a character they were working for better reference. The search for a simple and ubiquitous method to capture this acting motion led to the mouse and stylus as an input for hand movement which in turn lent to the premise of drawing. The assumption was that even a basic recording of human motion was able to express nuance and subtlety, so long as it was applied appropriately. This idea was first explored with a plugin developed for Autodesk Maya during a postgraduate project. The initial setup used Maya’s viewport for setting poses and previewing animation and separate window to layout keypoints and draw strokes.
Representing and Blending Keypoints

The idea for keypoints comes from the direct extrapolation of the keyframe in contemporary animation software. Since interpolation is now represented as a 2D drawing, the 1D keyframe must be turned into a 2D keypoint in order to interact with it. The radial basis function (RBF) of a Gaussian or 'bell' curve was used to describe the keypoint’s area of influence and for handling interpolation. The highest point on a gaussian curve is at its center and falls off with distance. This height determines the strength of the keypoint’s influence. Blending in between keypoints is done by overlapping their areas of influence and summing their respective heights. A RBF interpolation technique is used to ensure that keypoint influences do not exceed their expected values when added. The total collection of keypoint influences are represented as a heat map of sorts within the plugin window.
Recording and Manipulating Strokes

Strokes can be recorded by pressing the 'Draw' button and drawing on the plugin window. An internal timer keeps tracks of the time and creates a point at the cursor 25 times a second (25fps). The stroke was shown as simple line segments the connected the set of points.

A stroke could also be edited by moving any of its points. A 'soft-selection-like' method based on laplacian mesh editing techniques such as by Sorkine et. al (2004) was implemented so that as one point moves, the entire shape would adjust itself to maintain its overall smoothness. In theory this would allow more intuitive shape adjustments of strokes.

Design Comments

Little design consideration was taken on the user interface at this point of development. Aside from the drawing window, a set of buttons allowed for basic operations such as clearing the map, or adjusting keypoint parameters. Keypoints were created by selecting the desired objects and pressing the 'Create Pose Point' button. This created a 'snapshot'
of the objects in question and their respective position, rotation and scale. However, this snapshot was 'hard coded' and could not be changed after a keypoint was created, making it difficult to iteratively test different animations.

The Gaussian curve's relatively steep falloff curve results in an abrupt change in poses if the width of the keypoint is too narrow. While increasing the size of keypoint creates a gentler transition, it also takes up valuable space and may unintentionally affect the animation. This issue is compounded when interpolating multiple keypoints. Since an overlap is needed to blend smoothly, the further they are spaced apart, the larger they need to be. This takes up unnecessary space in the drawing window.

![Diagram of keypoint size and overlap](image)

**FIGURE 5.1.C. ILLUSTRATION OF KEY POINT SIZE AND OVERLAP AS THE DISTANCE INCREASES**

The presentation of the heat map was helpful in organising keypoints. However, since the colour scheme was the same for all keypoints, it was not possible to discern where the influence of one point starts, and another ends. Similarly, the overall shape of strokes had some interesting visual qualities; repetitive movements are shown as circular loops for example. But overlapping lines and the basic rendering style made it difficult to read the progression, speed and timing of the line.

The stroke editing functionality did not appear to be very useful overall. The implementation often led to undesired distortion of the stroke that needed further adjustments to correct. Additionally, the overlapping strokes made it difficult to figure
out what point was being adjusted. In many cases, I found that redrawing the stroke or changing the keypoint influences to be quicker and more intuitive.

![Figure 5.1.D. Plugin window showing unexpected deformation during stroke editing](image)

**FIGURE 5.1.D. PLUGIN WINDOW SHOWING UNEXPECTED DEFORMATION DURING STROKE EDITING**

One particularly useful piece of functionality was the 'Test' button, which did not draw any strokes by still showed the computed animation as the cursor was moved within the drawing window. This was surprising enjoyable as variations of the animation and speed can be tested in a non-destructive way.

Being able to utilise Maya's native viewport as well as other built in program functionality allowed for fast iteration and testing. That being said, there were many limitations to this approach. Having to contend with Maya's API without much experience in it meant that many workarounds had to made to construct this system. This caused unnecessary pauses and slowing in speed during certain operations, in particular when interpolation was calculated. Furthermore, an installation of Maya was needed to run the plugin, which is inconvenient for distribution and further testing.
5.2 Standalone Software Prototype

*FIGURE 5.2.A. THE INITIAL PROCESSING2 PROTOTYPE*

Following the proof of concept, I sought to build a more optimised and better designed version. Processing2 was used as the development platform for its graphics capabilities and personal familiarity.

**Keylines and Keyshapes**

From using the tool, I’ve found that there are 3 general use cases when interacting with keypoints:

1. Drawing from a keypoint from nothing or vice versa.
2. Drawing in between 2 keypoints.
3. Drawing in between 3 or more keypoints.

The initial proof of concept showed that creating a smooth interpolation between two keypoints takes up a lot of space when they are far apart. For the second case, this creates a lot of wasted space as the typical case would be to draw a more or less straight line directly between points. For the third case, manually adjusting the blending between 3 or points or more becomes impractical and difficult.
Keylines and keyshapes are introduced to address these cases. Connecting two keypoints creates an area of influence that joins the points together. Drawing along this keyline produces a smooth linear blend between the two points. Keyshapes are created by connecting 3 keypoints and perform a similar function. The triangle area enclosed by the keyshape creates a smooth blend of its respective keypoints based on the position within the triangle. Multiple keyshapes can be joined together to form more complex shapes.

**Development of Visual Style**

My original thesis proposal concentrated on the aesthetics of calligraphy as a major component of the development of this tool. As such, during development, I undertook Chinese calligraphy classes at inkLink Art Studio, Wellington, to better understand the subject matter. While my thesis objectives have since changed, much of the visual aesthetic was influenced by these initial classes.
The calligraphy ink brush is very sensitive to hand movements and is able to render these changes perceptively on paper through opacity and thickness. While in calligraphy, many factors come into play such as pressure, angle, speed and direction that are inherent to the brush, my software’s main concern is with the capture of speed. The distance between points is used to modulate the visual style of the stroke. Places where points are closer together are assumed to be slower and are made thicker and darker. Points spaced far apart are joined with lines that are thinner and lighter in colour. The distances were weighted logarithmically such that the stroke’s style is more sensitive to fast movement than it is to slow. This appears to give a more pleasing and consistent look.

Keypoints, keylines and keyshapes have also been given a new visual treatment. The addition of colours now allows individual keypoints to be identified, and interpolation of both keylines and keyshapes are visually shown as a blend of colours.
ANISKETCH: ALTERNATIVE AESTHETICS FOR COMPUTER ANIMATION TOOLS

FIGURE 5.2.D. STYLISATION OF A KEYLINE AND KEYSHAPE IN THE PROTOTYPE

2D vs. 3D Objects
As this new implementation was a standalone program, there was a debate as to whether to retain a 3D objects as in the plugin, or use 2D objects instead. The initial intent was to use 2D objects for simplicity. However, this was changed early in development in favour of a 3D objects in order to demonstrate and test more complicated animation examples. A basic 3D viewport with navigation, selection and basic primitives was built and utilised for this prototype. Objects supported basic operations such as translation, rotation, non-uniform scale as well as parenting and pivot adjustments to make more complicated characters.

User Interface
The user interface in this iteration was simplified, so as to not distract from the drawing process. The program window was split into two areas, for drawing and objects
respectively. In the middle were a column of buttons for creating poses, keypoints, 3D objects, and for testing, recording and playing strokes.

Contextual buttons for actions such as deleting, connecting or parenting are shown only if an applicable object is currently selected to avoid cluttering the interface. For actions that involved 2 objects, such as connecting keypoints, the button is dragged from the first object and dropped onto the second to indicate the concerning objects. Moving, rotating and scaling for 3D objects and keypoints are supplemented with additional gizmos.

Whenever a pose is created, it is now stored on a 'shelf' area on the right-hand side of the window. This allows poses to be created beforehand and reused multiple times. Selecting a keypoint opens a similar shelf where poses can be dragged and dropped into, associating the pose with the keypoint.

**Design Comments**

Keylines and keyshapes allow for a more efficient organisation of keypoints. More importantly, in addition to colours, these can be combined to produce visually striking layouts that may be reflective of the animation.

Being able to see the stroke thickness and opacity change while drawing provided a more aesthetically pleasing feedback and allowed users to respond by adjusting their speed accordingly. While the new style is more representative of its motion, readability issues caused by overlapping still persist, especially where the strokes are thicker and darker. Another issue was the natural shakiness of the hand, as creates a rough and noisy appearance at slower speeds.

A major issue that was encountered during the development of this prototype was the use of absolute position, rotation and scale for poses. While this did not affect animating singular objects such as a bouncing ball, it posed problems for layered animations. Consider a retracted arm made out of 3 parts for the upper and lower arm, and hand. Two poses are created: one of the arm extended, and another with the hand waving but
still retracted. It would make sense combining the two poses will make the arm extend and wave. Due to the way pose information is stored however, combining the two results in a half-waving, half-extended arm animation.

5.3 Final Design

The Processing2 prototype was useful in identifying critical issues and defining the base functionality and interface needed to implement this type of animation system. The final design presented in this paper, AniSketch, extends and refines on the concepts from the last prototype. The entire codebase on AniSketch was rewritten in eclipse, and the newer Processing3 library continues to be used as the graphics backbone of the software. The development for this version of the software was focused on creating a tool that could easily illustrate the core concepts and aesthetics of this type of animation system. For details on acquiring, running and using AniSketch, please refer to the accompanying documentation.

**FIGURE 5.3.A. ANISKETCH IN USE**
A Return to 2D Objects

While a 3D viewport would be able show more complicated examples, the choice was made to switch back to 2D objects for a number of important reasons. Firstly, navigating and coordinating objects in 3D is much more complicated and impedes the discussion and demonstration of the tool. Second, is that the extra data and calculations needed to handle the one extra dimension is more complex to build and requires more resources to run. Last, is that 2D is more accessible as a medium than 3D and creating or finding images to animate is much easier than doing the same with 3D models.

All 2D objects in AniSketch are represented as rectangles and support similar operations to the objects in the Processing2 prototype. Instead of allowing non-uniform scaling about the object’s pivot point, the width and height are allowed to freely change by manipulating the corners of the object. The ability to load images on top of objects was also added to allow more visually interesting creations.

Updated Keypoints and Relative Poses

Learning from the Processing2 Prototype, an alternative method for storing and assigning poses was developed. The original concept for a pose shelf with individual poses was scrapped in favour of a more elegant solution; keypoints and poses are combined as a singular component. Keypoints can be ‘opened’, at which point any relative changes made to the objects in the scene are recorded and stored. This is different from the earlier versions where the absolute values position, rotation and scale are stored. When poses are applied, only objects with said relative changes are affected. This method allows for layered animations.

Improved Stroke Styling and Functionality

Solutions to both issues of visual noisiness and overlapping were attempted. To mitigate hand jittering, stroke thickness and opacity at any point are now based on the average distance to its 10 adjacent neighbours, providing an overall smoother look. The problem of overlapping was addressed by fading away strokes that were too far away in the past or future from the current point in time. This removes strokes that not in use.
Additionally, the overall visual aesthetic of the stroke has been adjusted to give a more ink-like feel. Drawn strokes are now much lighter in colour as though drawn with a wet inkless brush. Points on the stroke that are of interest, such as the point currently being traversed or that is closest to the cursor, are highlighted to look like a drop of ink spreading out along the line. This new style is not as striking as the previous prototype, but offers better clarity.

The functionality of strokes have also been extended to support different animation behaviors. Typically, a stroke is interpreted by moving along its points until it reaches the end where the stroke is no longer interpreted. Now in addition, stroke behaviours can be set to 'hold', where the last point continues to be interpreted, or 'loop' where the stroke repeats itself.

**Gestures**

Where applicable, AniSketch supports the use of simple contextual gestures in place of buttons to perform functions and edit objects. Gesture recognition allowed for a cleaner interface and more space for drawing. Gestures are executed by drawing while holding down right-mouse or the stylus button over the object or area concerned. The gesture recognition system used in AniSketch is based on the $1$ Recognizer algorithm by Wobbrock, Wilson, & Li (2007). $1$ was primarily chosen for its ease of implementation, reliability and speed. Limitations cited by the authors in rotation invariance and 1D lines have been addressed with additional stroke processing where necessary. The gesture language used was inspired by SILK by Landay & Myers (1995).

**User Interface**

AniSketch retains the same general layout from the previous prototype. A timeline area was added to bottom of the window for playback functionality. All areas can be resized to provide more space where needed.
6. Animation Examples

This section contains some examples that illustrate the animation concepts available to AniSketch. Each example will contain a graphical notation of the entire composition and its components. An accompanying screenshot with the composition assembled in AniSketch is provided for reference. The following diagram describes the notations used in these examples:

![Diagram of AniSketch graphical notation legend]

**FIGURE 6.A. ANISKETCH GRAPHICAL NOTATION LEGEND**

Please note that the animation notations are only a general indicator of how to create these compositions and should be actively deviated from. The same animation may be able to be created in a different way. For video footage of the animations in these sections and more animation examples, please refer to the accompanying materials.
6.1 Yet Another Bouncing Ball

**FIGURE 6.1.A. ANISKETCH NOTATION FOR A BOUNCING BALL**
FIGURE 6.1.B. ASSEMBLED ANIMATION FOR A BOUNCING BALL IN ANISKETCH
6.2 Walking An Elephant

**FIGURE 6.2.A. ANISKETCH NOTATION FOR A WALKING ELEPHANT**
**FIGURE 6.2.B.** ASSEMBLED ANIMATION FOR A WALKING ELEPHANT IN ANISKETCH
6.3 All Seeing Eye

**FIGURE 6.3.A. ANISKETCH NOTATION FOR A MOVING EYE**
FIGURE 6.3.B. ASSEMBLED ANIMATION FOR A MOVING EYE IN ANISKETCH
7. Discussion

7.1 The Aesthetics of AniSketch

No formal user testing was undertaken for AniSketch. One of the main objectives of this paper is to design an animation system that is in reaction to the current established modes of animation production. As such, as my understanding of the subject matter progressed, so did the software’s functionality and purpose continue to change. Undergoing various states of gestation and instability, it was difficult to find an opportune time for formal testing. Returning to *The Three Paradigms of Human-Computer Interaction* by Harrison et al. (2007), the authors assert that objective measures alone only provide at best, an incomplete assessment. "Instead, we must ask questions about what it means for a system to be 'good' in a particular context" (p. 8).

My discussion for this paper focuses on establishing the aforementioned context through examining the aesthetics of AniSketch in relation to existing computer animation tools. These aesthetic qualities are derived from observations that are inherent to AniSketch’s modes of interaction and the computational techniques that underlie it. In my attempt to describe the essential nature of the tool, I hope to establish a useful framework to support future discussions and assessment of AniSketch as well as other alternative animation systems.

I have chosen to group the aesthetics of AniSketch into 2 types of user experiences: the evocative and provocative experience.

**Evocative Experience**

*In The Language of New Media*, Manovich (2001) describes digital media as possessing 'variability' and 'modularity' (p. 30-45), wherein a piece of digital media is composed of smaller essential components that can be independently remixed to produce unique combinations. In AniSketch, every component of the output animation—from the keypoints and the contained pose data, to the nature of the motion in the drawn strokes—is made visually apparent and accessible to the animator. This is in contrast to
conventional animation tools where the underlying data and computations are presented as disassociated curves and numbers that do not make their relationship to the animation immediately clear. By exposing the imagery of animation's structure and data, there is a freedom for it to shape and be shaped by the artist's own perception, enabling an *evocative experience* of the animation through its essential components.

AniSketch’s use of a sheet onto which movements are drawn is essential to supporting this experience. The decoupling of movement (drawn strokes) and image (objects and poses) allows the ability to reuse the motion in a variety of contexts. The drawing serves to convert the natural modalities of the hand into discrete points that are displayed in a way that accentuates the nature of its movement. Through this interface, the artist’s own drawing style and notation can be appropriated, together with drawn shapes and curves, into a visual lexicon of motion, and a faint semblance of an artistic mark.

Keypoints and strokes come together to form visual compositions that control the animation output. This 2D composition is markedly different that the 1D timeline that is familiar to animators. With an extra dimension, there is more freedom to move and rearrange. What was a single direction of time, is now omni-directional. From this freedom, comes variability. 2 animations may look the same on the outside, but its underlying composition can be completely different.

The display of keypoints, keylines and keyshapes are visual representations that reveal the computational behaviour of AniSketch, allowing for the crafting of control interfaces. The structure of these arrangements can be made to be reflective of the animation, or orthogonal to it to incite an emotional response and augment the resulting movement. Visual compositions can be used to document and communicate an artist's own internal perception and motives.

**Provocative Experience**

Provocative experience refers to AniSketch's ability to induce agency, spontaneous action and user reflection through its modes of interaction and computation. Production-orientatated animation tools are inherently about precision. Keyframes store a
perfect record of an object's attributes at a point in time, enabling computer automated interpolation to reliably and predictably reproduce animations. In AniSketch, nothing is numerically indicated to the user and no facility is given for precise adjustments. In contrast, the keypoints in AniSketch only store relative changes, providing an incomplete snapshot of an object's state. Because of this, it is not immediately clear what would result when combining multiple keypoints together. The only way to understand their behaviour is to actively create animations through drawing. While the user is drawing, the resulting animation is played back in real-time, allowing the user react to the motion and make adjustments accordingly. This mirrors Lasseter's description of the animation technique of straight ahead:

‘He knows where the scene fits in the story and the business it has to include. He does one drawing after another, getting new ideas as he goes along, until he reaches the end of the scene. This process usually produces drawings and action that have a fresh and slightly zany look, because the whole process was kept very creative.’ (Lasseter, 1987, p. 40)

The natural deviations of drawing mean that a perfect reproduction of the animator's intent is practically impossible. The mathematical qualities of the Gaussian curve ensures that keypoints always exert some influence no matter how far they are, and that no pose can be perfectly recreated. Furthermore, drawn strokes cannot be edited after the fact and have to be redrawn if the result is not desired. This real-time, imperfect reproduction of the artist's intent provokes a continuous assessment of the subtle perturbations, unintended influences and mistakes brought about through the animation process—at the very least from a poetic standpoint. As Jones (1999) puts, 'You must respect the impulsive thought and try to implement it. You cannot perform as a director by what you already know; you must depend on the flash of inspiration that you do not expect and do not already know'.
7.2 Potential Applications

The aesthetics of AniSketch animation system make a suitable informal animation tool for creative exploration. In a preliminary study by Davis & Landay (2008), the authors found that animators expressed interest in the use of more informal animation tools as prototyping aids. The interviews also found that many non-animators had envisioned uses for animation tools, but did not want to spend effort learning a complex tool. Example uses cited by non-animators included illustration and explanation of concepts, for entertainment, as a teaching aid and for choreography planning (p. 414).

Another application of AniSketch is for recreational use, or as an interactive animation 'toy'. Phantomanation (2012) is a videogame that teaches players basic animation concepts. A promotional website for the band OKGO, iwontletyoudown.com (n.d.) features an online animation tool that recreating the band’s dance routines.

Since drawing and visual composition are key elements of the animation process, creating an animation becomes a form of visual performance art. AniSketch could be used as a component for live performances such as VJing, or as interactive storytelling tools for audiences.

7.3 Limitations and Future Work

One major challenge with AniSketch’s animation system is space. The small sheet size means that sketches and compositions are limited in their complexity. A possible way to address this is through a navigation system to the allow the user to move around the sheet, or zoom out for a better overview. Another solution is to introduce a layer system.

As compositions become more complex, it is not clear if users are able to keep track of the components in the scene. Adding an annotation layer for writing and sketching comments could be useful for organisation and planning.
Overlapping strokes continue to be a problem by making it difficult to read the composition. An interesting way of tackling this issue could be trying to 'expand' the stroke's shape in a way that shows more detail. Imagine the stroke like a spring being stretched out into a long wire. After edits have been made, the stroke can collapse into its original shape.

While adequate, the gestures in AniSketch are occasionally misrecognized or not recognised at all. One reason for this is because the basic gestures used to train the recogniser are not customised for the user. A more user friendly interface could show the best guesses whenever a gesture is poorly recognised. The user can then select the appropriate gesture to train the system. A more robust recognition system can also help improve the gesture experience.

Aside from deleting, strokes cannot be edited or adjusted in any other way. In future versions of AniSketch, I would like to explore how strokes could be edited and made to interact with other components in the composition. Potential ideas include soft-selection editing, erasing, splitting or joining strokes, and applying filters to change its behaviour.

As of writing this thesis, only a few animation examples have been explored and presented with AniSketch. I fully intend to distribute AniSketch to relevant groups as well as carry out more formal user testing in order to understand the full range of capabilities and limitations of the tool. I believe that this is essential to determine future design and development directions for the software. To support this, additional feedback mechanisms could be added to facilitate testing. In addition, I plan to continue to use AniSketch to create more complex examples and full animations to test its viability.

Another future implementation of the AniSketch system could be as a plugin or subsystem in a professional animation package. While AniSketch is meant as an informal tool, it would be interesting to see how such a system can be used to augment the
professional animation pipeline and likewise how AniSketch can leverage the ecosystems provided by professional software.

8. Conclusion

What are the aesthetics of computer animation?

On reflection, this is difficult to answer with certainty. While there is certainly a causality between the animation process and tools used, one cannot discount other influential factors such as cost, or studio politics. That being said, a few outlooks can be given on the place and perception of contemporary computer animation tools.

It is clear that computer animation tools borrow from the modes of production defined by divisions of labour. Some elements, such as keyframe interpolation were sought to replace their human counterparts. But others, such as the subscription to photorealism, the use of digital 'bones' and the obfuscation of data can also be seen as a perpetuation of Disneyan proclivities—one that is focused on the efficient reproduction of naturalistic movement. In his book, Algorithmic Form, Terzidis (2003) describes this type of 'computerization' of existing processes as inherently limiting, as the parameters of these digitised entities have already been pre-defined. (p.69)

However, this view perhaps does not account for the user's own perception and agency with regards to these tools. As Moszkowicz (2002) points out, 'the [animation] medium has always enjoyed creative relationships with new technology. Indeed, since its inception, practitioners have worked hard to establish animation both artistically and industrially, and an integral part of this project has been the development of a specific aesthetic language' (p. 296). While this relationship obviously plays a role in fanfare around the latest rendering techniques that seem to make the rounds year after year, there more subtle ways in which this manifests.
The perception of the computer and its role in animation evolved as technology progressed. From an animator's perspective, the computer's role has shifted from a novelty in the 70s and 80s (Tai, 2013), to aversion in the 00s as digital pipelines started to take over (Tarantini, 2012), to the somewhat empathetic understanding of its capabilities today. This speaks to the continued use of hand drawn storyboards and pencil tests. Likewise, the proliferation of plugins and extensions as well as their support from software developers signals an admittance of sorts to the software's inherent limitations the need for appropriation by users. From an industry perspective, 'younger' studios that were born in the digital age of animation, such as Pixar, recognise and attempt break away from divisions of labour that have come to characterise their older counterparts (Power, 2012, p. 36). From the point of view of computer graphics, the development of key rendering technologies in the 90s and 00s, also marked the birth of the many alternative animation tools, such as the field of sketch-based interfaces.

The assumption of the thesis statement that 'current user interfaces and techniques used by computer animation tools inherently impose aesthetic limitations on the artist' are only valid if the artist chooses to not engage with the greater discourse around these tools.

*And what are the benefits to building computer animation tools with alternative aesthetics?*

The development of different animation aesthetics is one of provocation. The shaping of the animation medium seems to often involve strife. Manovich (2001) paints the birth of animation arising from the tension between itself and cinema, describing it as 'cinema's bastard relative' (p. 298). Power (2012) notes that the style of limited animation was developed in reaction to Disney animators fed up over the studio’s naturalistic doctrine (p. 30). Jones, describes conflict as an important part of the creative process, 'creativity without opposition is like playing polo without a horse.'

Aside from research, there are not many tools that strive to be different from the current status quo. Many of these tools subscribe the same aesthetics from a legacy shaped by
decades of research. Left without provocation, the stagnation of aesthetics, tools and features are bound to continue, held steady by its inertia.

The aesthetics of Anisketch lie in its agency to upset this inertia. The provocative and evocative nature of the software constantly challenges the animator at work, provoking new ideas and new combinations. AniSketch does not mediate the user experience, nor does it hide its inherent aesthetics. There is no automation—using this tool is literally invoking life.

Is is through new aesthetics that we can put established ones into context. Like the conflict between tool and the user, it is only through the conflict of tools that we come to fully understand and value it. It is only inevitable from a medium whose working central principles are defined by words such as such as 'extremes', 'exaggeration', and 'anticipation'. The very nature of animation depends on it.

9. References


doi:10.1145/258734.258902


doi:10.1145/800248.807414


doi:10.1145/1294211.1294238

Fischinger, O. (Director). (1931). *Study No. 8* [Short Film]. Germany: Fischinger Studio.
