Designing Digital Art and Communication Tools Inspired by Traditional Craft

Kazi Rubaiat Habib
(B.Sc(Hons), BUET)
Computer Science, School of Computing
National University of Singapore

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
COMPUTER SCIENCE, SCOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2014

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Kazi Rubaiat Habib
5 May 2014

ACKNOWLEDGEMENTS

I would like to thank my academic supervisor Dr. Shengdong Zhao for contributing in every aspects of my academic endeavor. He encouraged me to conduct original research with high standards and pursue my true vision. I was trained to develop myself as an independent researcher from the very beginning. He also introduced me to many notable researchers, which helped me to widen my vision and develop invaluable collaborations. I am also very grateful to him for all the financial and psychological assistance provided during the physical and mental trauma I have encountered throughout the PhD period. I have learned so much from his commitment to work, critical thinking, time management and decency, which will have far-reaching impact on my life.

I thank my parents Dr. Kazi Belayet Ali and Nazneen Ali for all the supports and encouragements, which greatly motivate me to fully concentrate on my research. I am also very grateful to my beloved wife, Sarah Tasnia, for her continuous encouragement and patience during the period.

During the course of my PhD, I had to pleasure to work and collaborate with many other reputable researchers in my field. I would like to extend my sincere thanks and gratitude to my collaborators Richard Davis (Singapore Management University), Takeo Igarashi (University of Tokyo), Low Kok-Lim (National University of Singapore), Bongshin Lee (Microsoft Research), Yin Kangkang (National University of Singapore), Tovi Grossman (Autodesk Research), George Fitzmaurice (Autodesk Research), Fanny Chevalier (INRIA), Chris Chua (ThousandEyes) and many others from whom I have learned so much. Working with all these researchers helped me to widen my vision, thinking and perspective.

I would also like to thank to all my colleagues, friends and fellow graduate students in NUS-HCI Lab. I have learned a lot from my peers. I also received invaluable feedback, insight and comments from my peers in my research.

I have received support in the form of scholarships, facilities or research fund from National University of Singapore, Microsoft Research (Redmond, US), Microsoft Research Asia (Beijing, China), Japan Science and Technology Agency (JST) and Autodesk Research (Toronto, Canada). This research would have been impossible without that generous support.

ABSTRACT

The goal of this work is to design and develop computational interfaces that connect users with to creative process with seamless user experience and enable powerful ways of content creation. This thesis presents the design and implementation of four digital art and communication systems **-** ***SandCanvas*** (Chapter 3), ***Vignette*** (Chapter 4), ***Draco*** (Chapter 5) and ***SketchStory*** (Chapter 6). ***SandCanvas*** is a digital multi-touch application for real-time storytelling in which an artist dexterously manipulates virtual sand to produce images and animations, inspired by sand animation. ***Vignette*** is a style-preserving sketching tool for pen-and-ink illustration with built-in texture synthesis capabilities, preserving traditional workflow and artistic styles by synthesizing from example strokes. ***Draco*** is a sketch-based interface that enables users to add rich set of animation effects consisting groups of strokes with co-ordination motion to their drawings, seemingly bringing illustrations to life. ***SketchStory*** is a data-enabled digital whiteboard, facilitating the creation of personalized and expressive data charts quickly and easily. ***SketchStory*** recognizes a small set of sketch gestures for chart invocation, and automatically completes charts by synthesizing the visuals from the presenter-provided example icon and binding them to the underlying data.

The design of these tools exemplifies how an in-depth and holistic understanding of traditional workflow, artifacts, medium and human needs aids and inspires the design and development of new digital new media art tools. These computational systems seek inspiration and design insights from traditional and existing art mediums, but the new computational affordances and seamless user experience affords entire new possibilities and a boundless space for possible outcomes. While preserving the style of the artist, art form and essential workflow of the creation process, these tools use the capabilities offered by digital technology to accelerate the tedious components of the original process by synthesizing from example sketches spatially (***Vignette***), spatio-temporally (***Draco***) or from data (***SketchStory***).

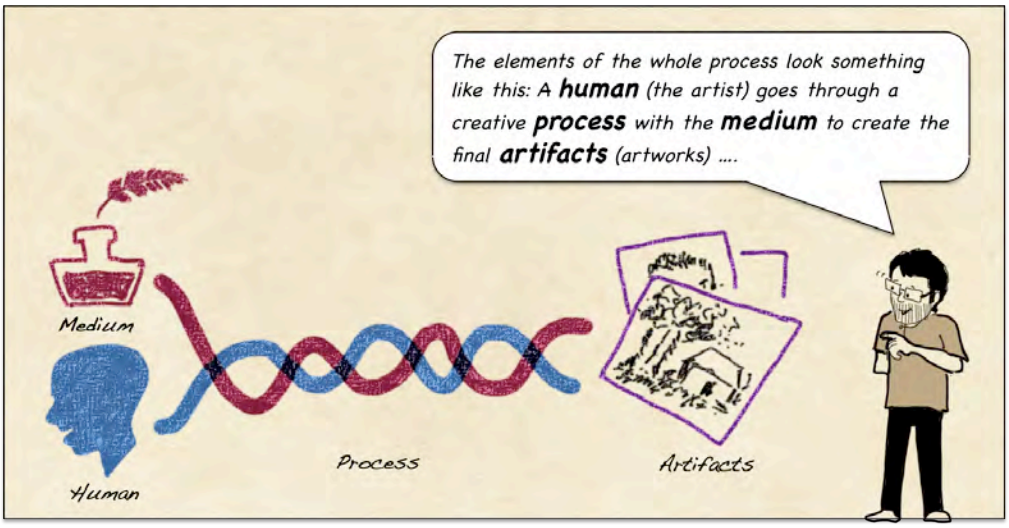
Table of Contents

List of Figures

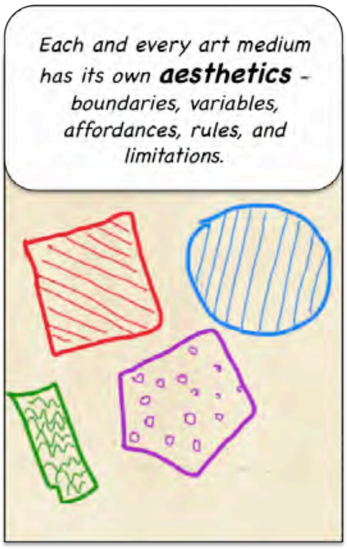
INTRODUCTION

1.1 Motivation and Goals

Over thousand of years, our ancestors have built sophisticated artwork practices, methods, and aesthetics.
How can we learn from this rich art history and existing practices?
Traditional art media (like watercolor, oil painting, etc.) offer an abundance of choices in raw materials, tools and techniques.
Painting tools and processes capitalize on the delicate mechanical structure of human hands and their affordances..
In contrast, expressive digital art tools rely on indirect manipulation of numerous parameters.
Ps
How can we mediate between existing practices and technology?
Technology
Practice
We can explore ways to mediate between traditional art and technology by studying an existing physical medium- it can be sand art, water color, ink illustration, or anything else.
The next step is to understand the medium. How does it work? How can the process be learned, discretized, and quantified? How can this understanding be converted into useful design knowledge?
To answer these questions, let's take a look at the big picture of the whole process -



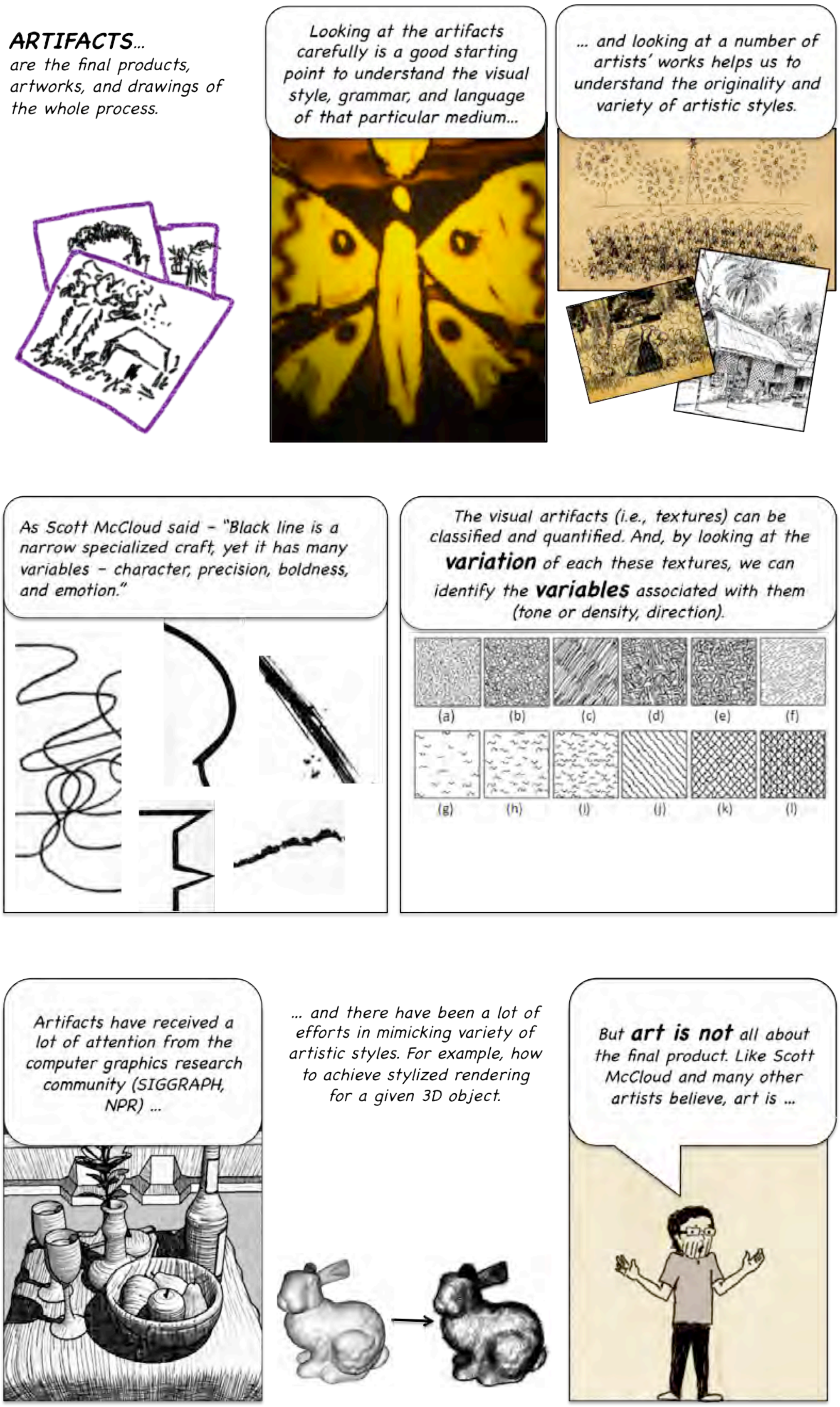
*There are a plethora of physical art **MEDIA** with distinct visual styles and creation process, (e.g., watercolors, ink illustration, sand art, pencil sketch, etc.)*



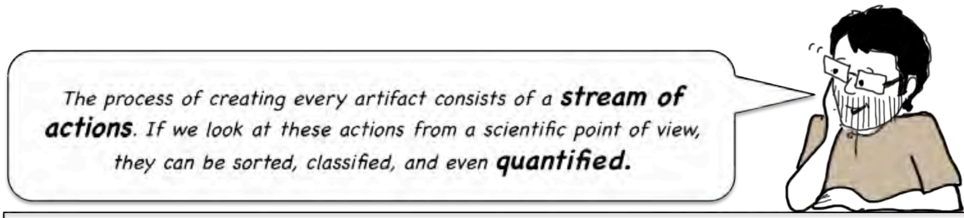
"At the heart of any craft practice, lies the idea of applied, skilled understanding and mastery of material…"
*- **Scott McCloud***

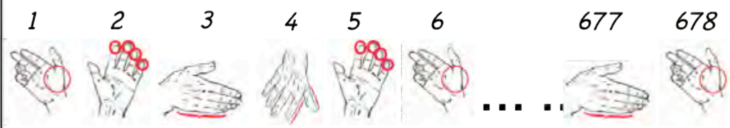
*The **HUMAN** counterpart demonstrates that understanding by creating the artifacts.*

*For instance, as a performance art medium, the beauty of sand animation lies not only on the graphic style or final drawings, but **how** those drawings are created in **real-time**. The elegance of sand animation lies in the seamless flow of expressive hand gestures that cause images to fluidly **evolve** from one image to another, surprising and delighting audiences. A skilled sand animator designs a story that leverages the essence of storytelling techniques in sand animation.*


ARTIFACTS...
are the final products, artworks, and drawings of the whole process.
Looking at the artifacts carefully is a good starting point to understand the visual style, grammar, and language of that particular medium...
... and looking at a number of artists' works helps us to understand the originality and variety of artistic styles.
As Scott McCloud said – "Black line is a narrow specialized craft, yet it has many variables – character, precision, boldness, and emotion."
The visual artifacts (i.e., textures) can be classified and quantified. And, by looking at the variation of each these textures, we can identify the variables associated with them (tone or density, direction).
(a) (b) (c) (d) (e) (f)
(g) (h) (i) (j) (k) (l)
Artifacts have received a lot of attention from the computer graphics research community (SIGGRAPH, NPR) ...
... and there have been a lot of efforts in mimicking variety of artistic styles. For example, how to achieve stylized rendering for a given 3D object.
But art is not all about the final product. Like Scott McCloud and many other artists believe, art is ...

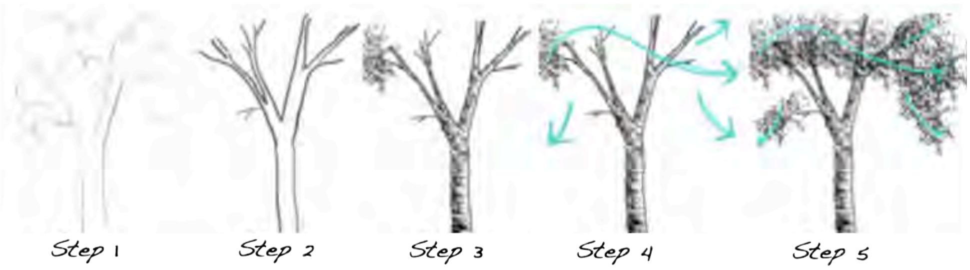
...the **PROCESS** of creation. Art is the creative interplay between the medium and the human.

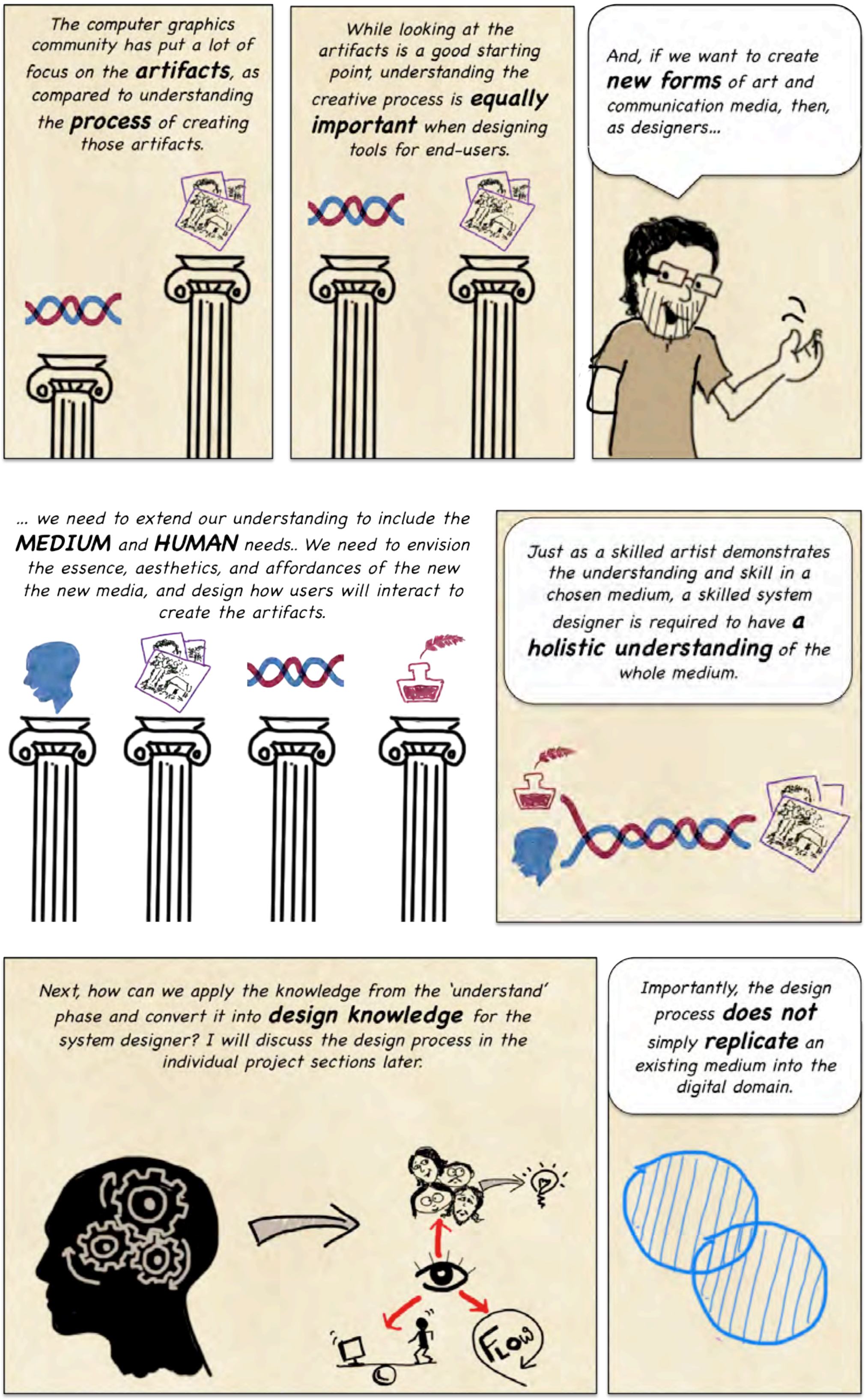




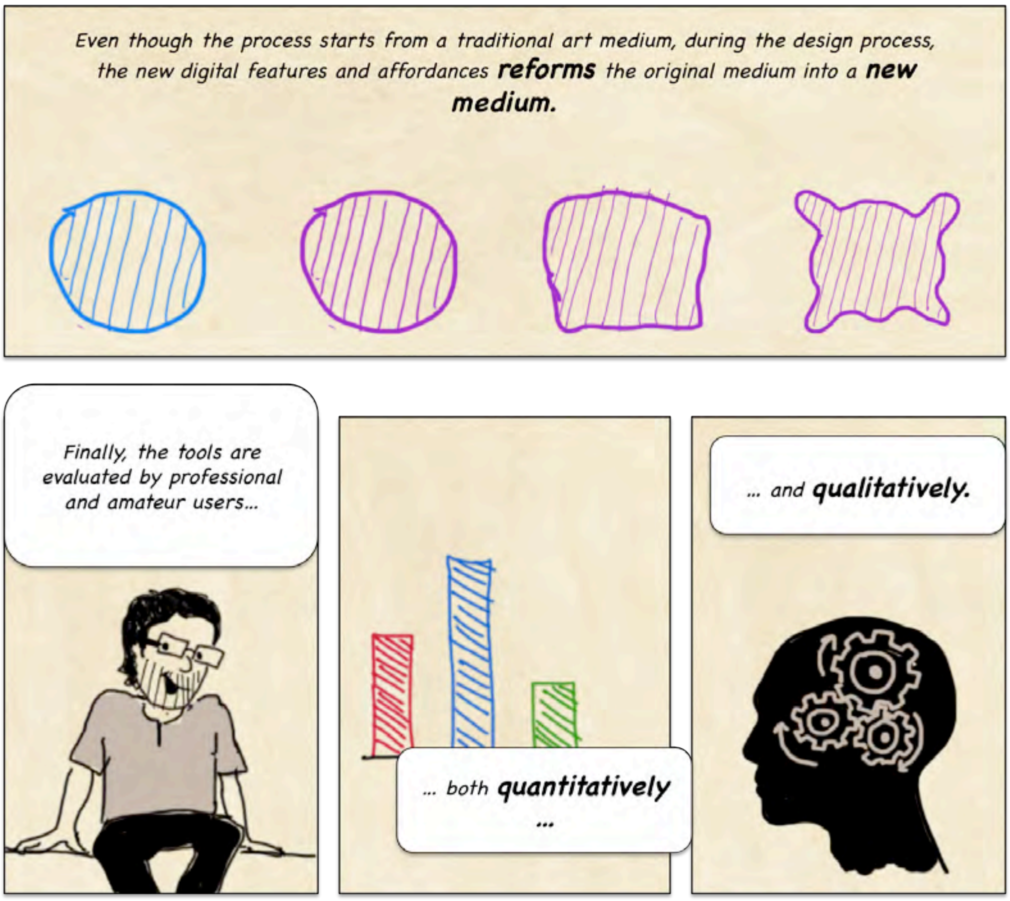
For example, a sand animation is created with a sequence of hand gestures using different parts of the hand.

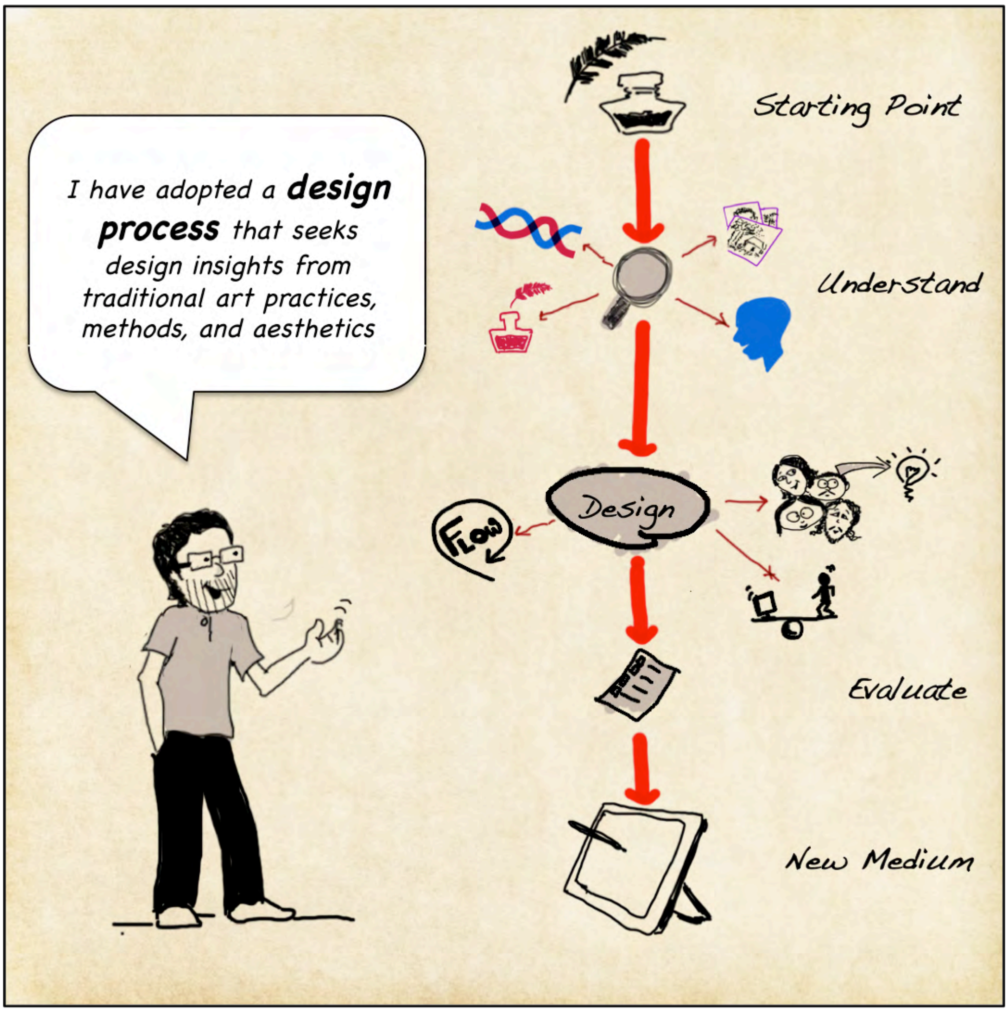
Looking at the steps of traditional workflow helps us to understand how an artist translates visual form (or thoughts) into structures and steps to create the final artifacts.

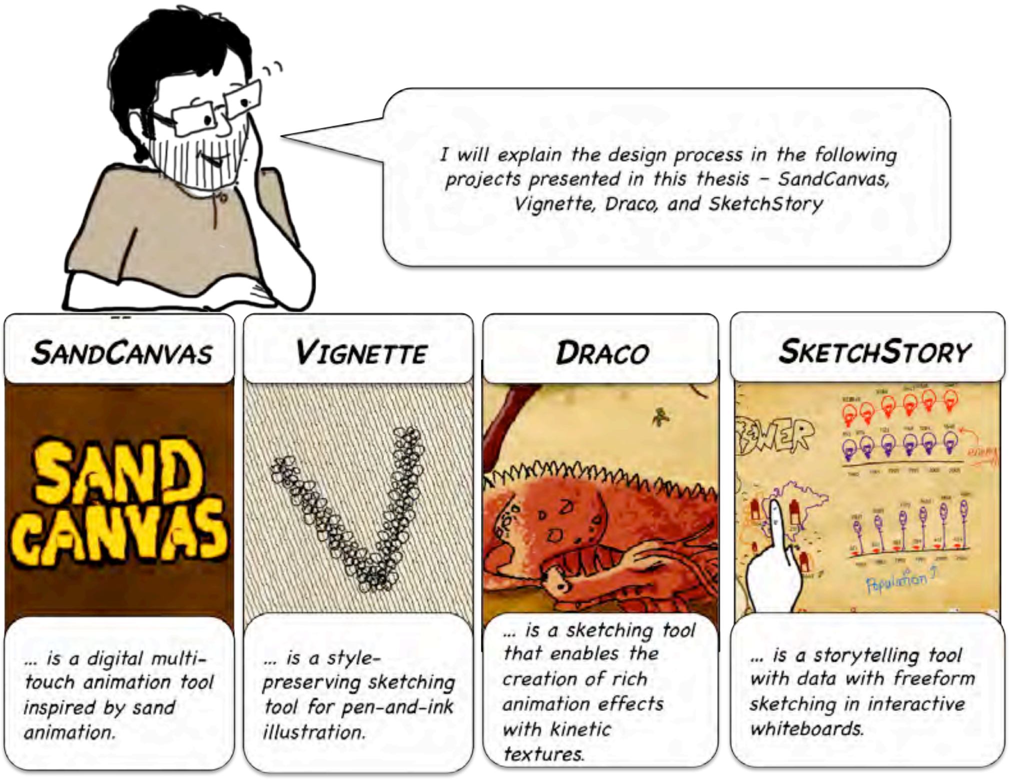




The computer graphics community has put a lot of focus on the **artifacts**, as compared to understanding the **process** of creating those artifacts.
While looking at the artifacts is a good starting point, understanding the creative process is **equally important** when designing tools for end-users.
And, if we want to create **new forms** of art and communication media, then, as designers...
... we need to extend our understanding to include the **MEDIUM** and **HUMAN** needs.. We need to envision the essence, aesthetics, and affordances of the new the new media, and design how users will interact to create the artifacts.
Just as a skilled artist demonstrates the understanding and skill in a chosen medium, a skilled system designer is required to have **a holistic understanding** of the whole medium.
Next, how can we apply the knowledge from the 'understand' phase and convert it into **design knowledge** for the system designer? I will discuss the design process in the individual project sections later.
FLOW
Importantly, the design process **does not** simply **replicate** an existing medium into the digital domain.

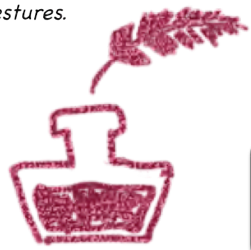

Even though the process starts from a traditional art medium, during the design process, the new digital features and affordances **reforms** the original medium into a **new medium.**
Finally, the tools are evaluated by professional and amateur users...
... both **quantitatively** ...
... and **qualitatively.**

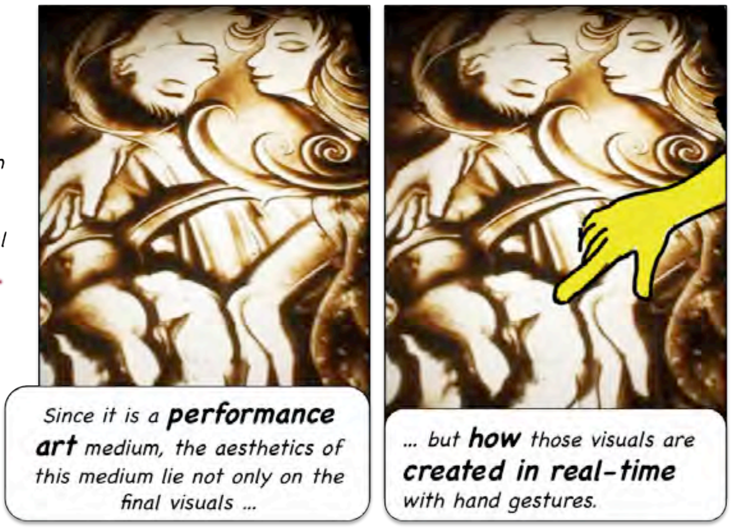

I have adopted a **design process** that seeks design insights from traditional art practices, methods, and aesthetics
Starting Point
Understand
Design
Flow
Evaluate
New Medium


I will explain the design process in the following projects presented in this thesis - SandCanvas, Vignette, Draco, and SketchStory
SANDCANVAS
SAND CANVAS
... is a digital multi-touch animation tool inspired by sand animation.
VIGNETTE
... is a style-preserving sketching tool for pen-and-ink illustration.
DRACO
... is a sketching tool that enables the creation of rich animation effects with kinetic textures.
SKETCHSTORY
... is a storytelling tool with data with freeform sketching in interactive whiteboards.

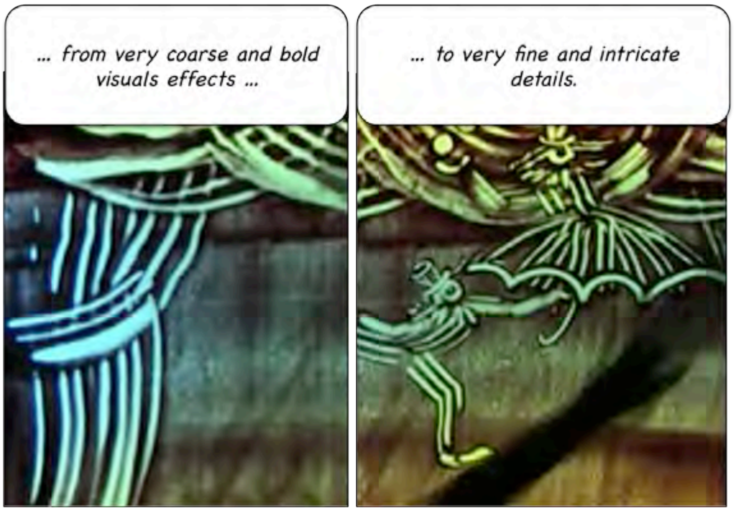
1.2 SandCanvas

Sand animation, as a **MEDIUM,** has enjoyed widespread popularity for its fluidity and unique graphic style. The artist unfolds a narrative through a progression of visual images produced with a seamless stream of physical gestures.

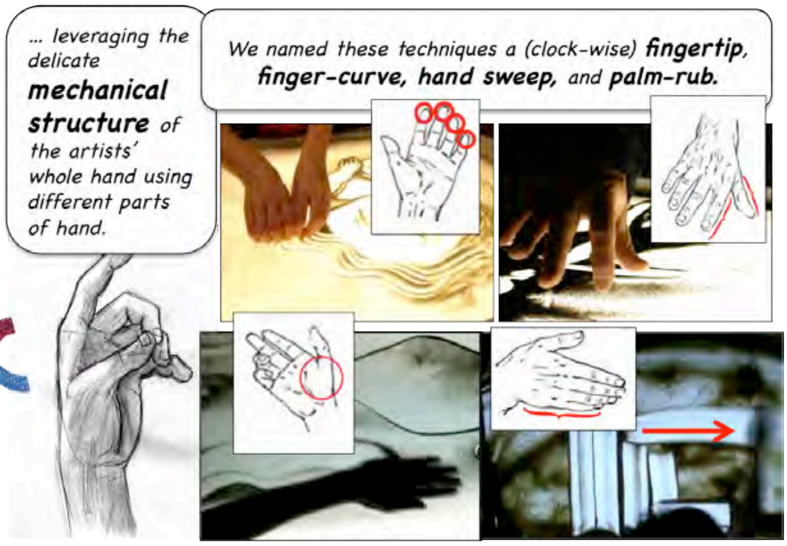


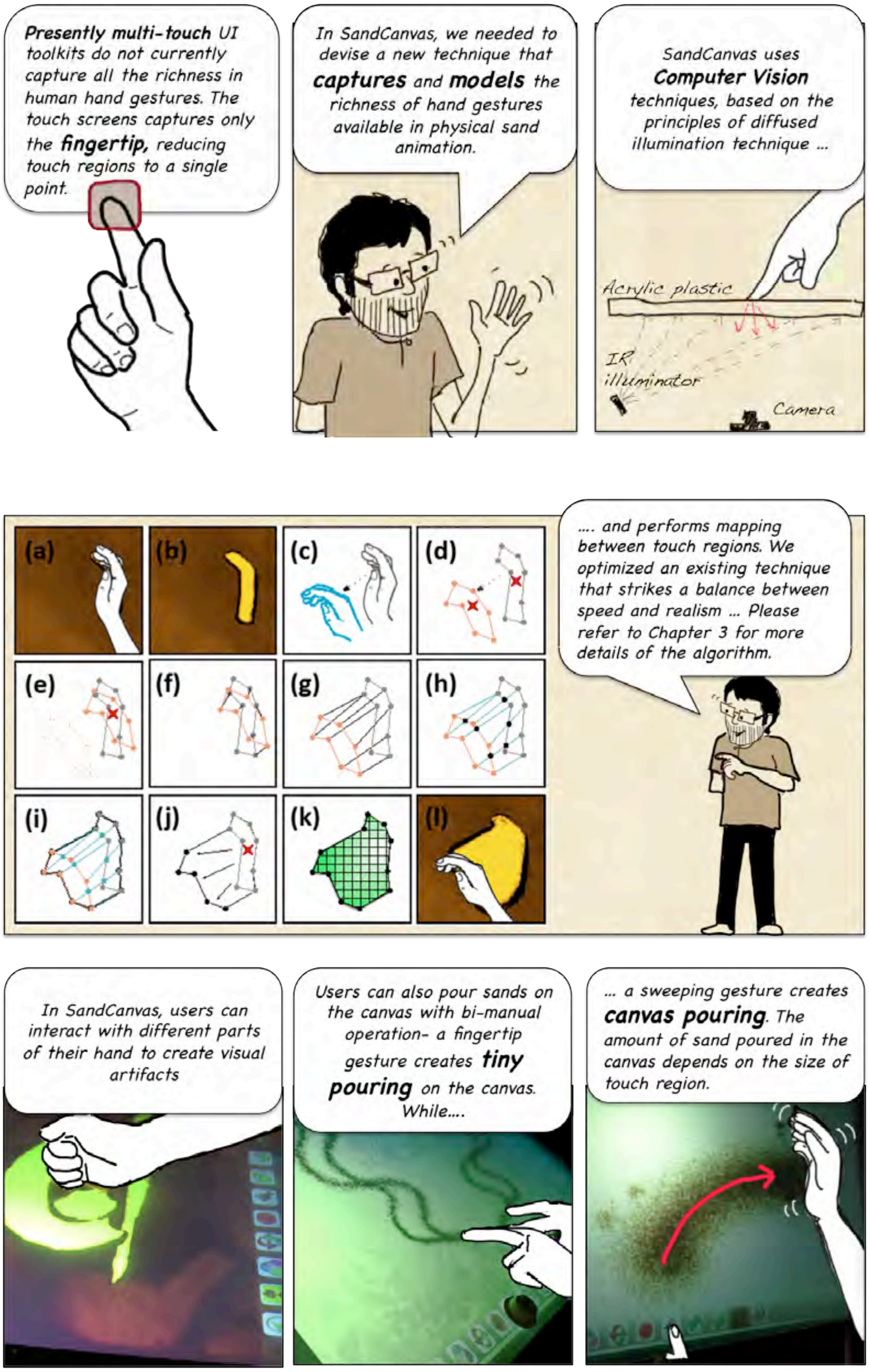


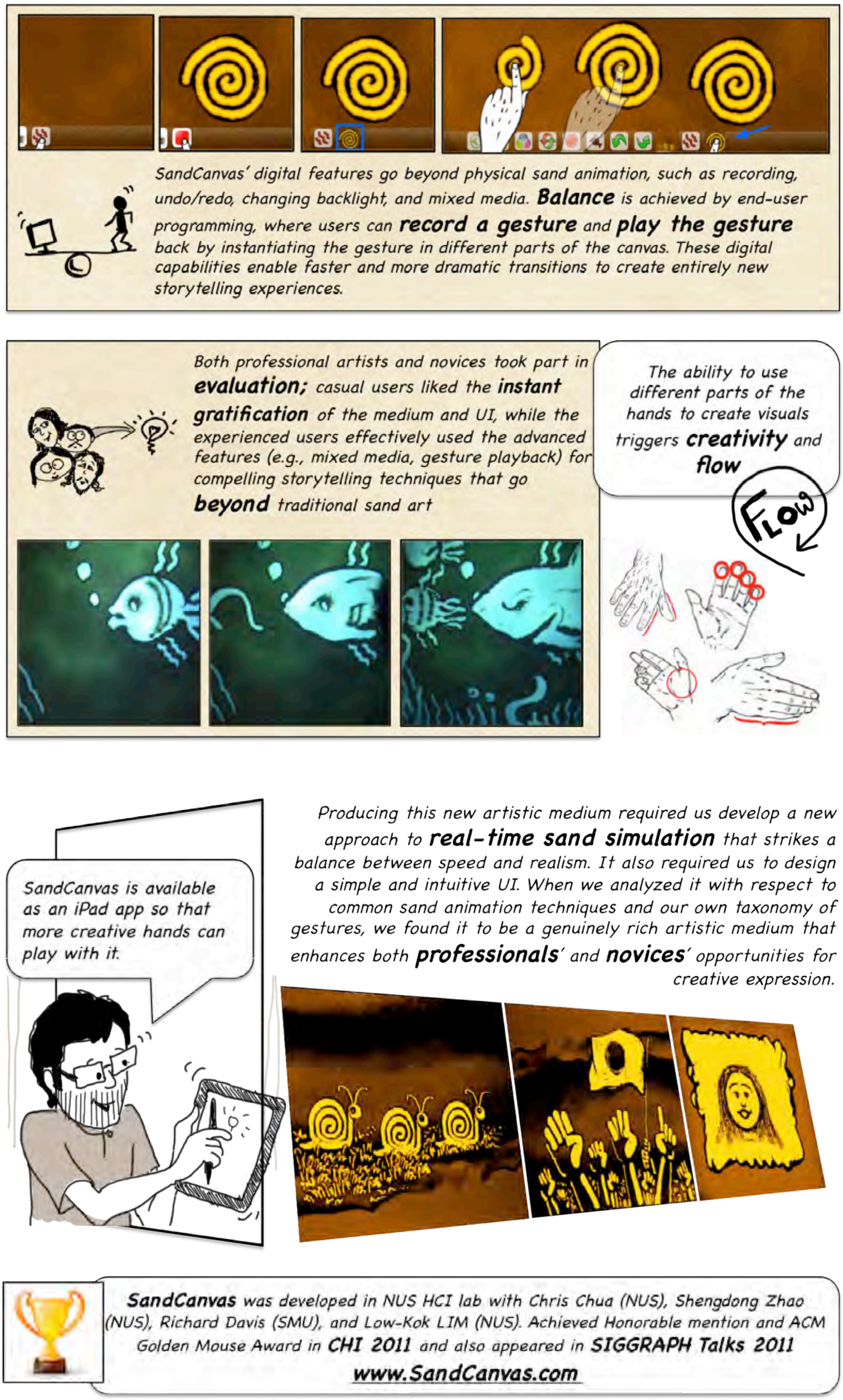
Looking at the **ARTIFACTS**, sand animation possesses a rich and wide range of visual effects, using of sand and hand only, ...



The **PROCESS** of creating those visuals consists of a stream of powerful and expressive vocabulary of hand gestures, ...

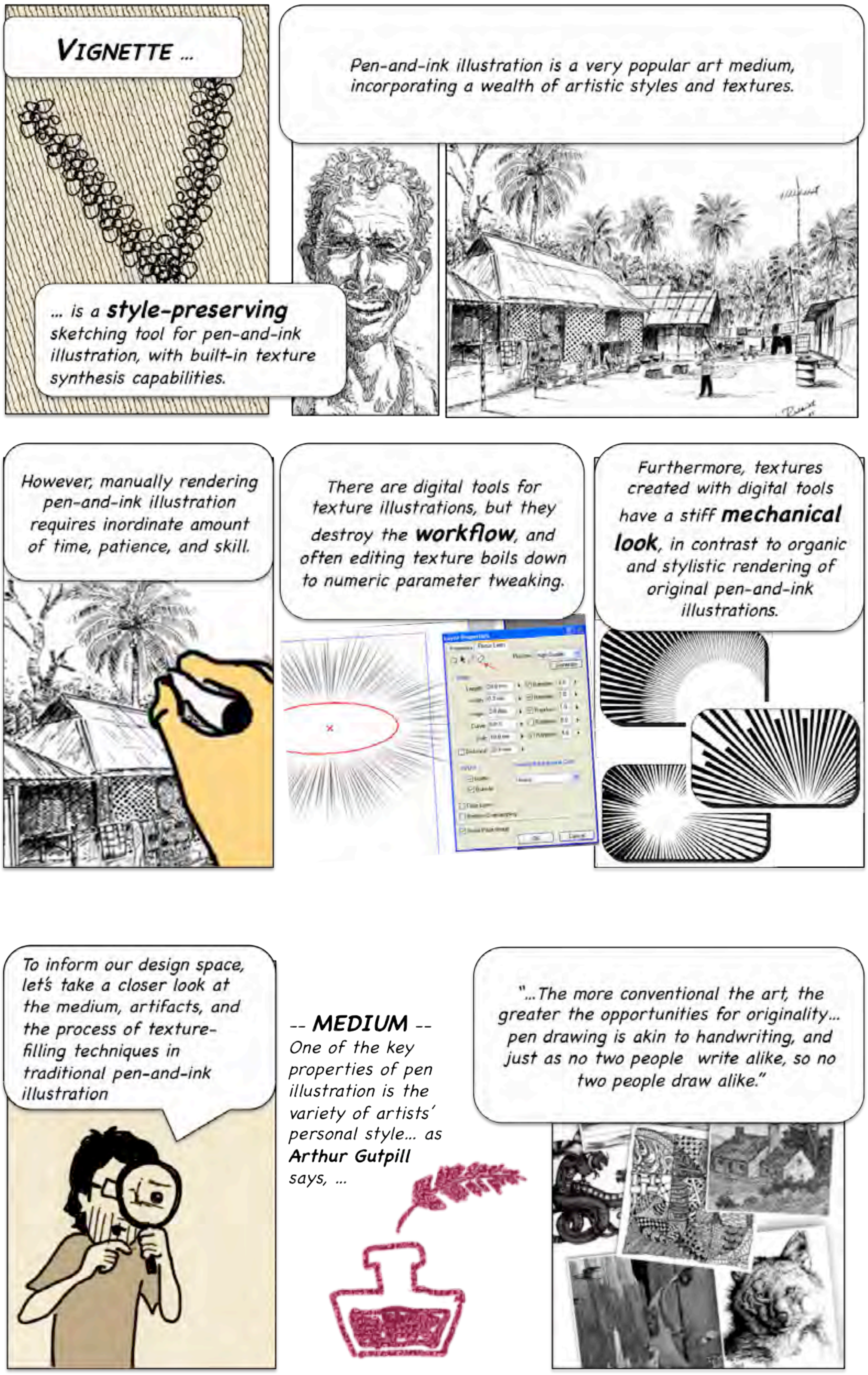



Presently multi-touch UI toolkits do not currently capture all the richness in human hand gestures. The touch screens captures only the fingertip, reducing touch regions to a single point.
In SandCanvas, we needed to devise a new technique that captures and models the richness of hand gestures available in physical sand animation.
SandCanvas uses Computer Vision techniques, based on the principles of diffused illumination technique ...
Acrylic plastic
IR illuminator
Camera
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)
.... and performs mapping between touch regions. We optimized an existing technique that strikes a balance between speed and realism ... Please refer to Chapter 3 for more details of the algorithm.
In SandCanvas, users can interact with different parts of their hand to create visual artifacts
Users can also pour sands on the canvas with bi-manual operation- a fingertip gesture creates tiny pouring on the canvas. While....
... a sweeping gesture creates canvas pouring. The amount of sand poured in the canvas depends on the size of touch region.



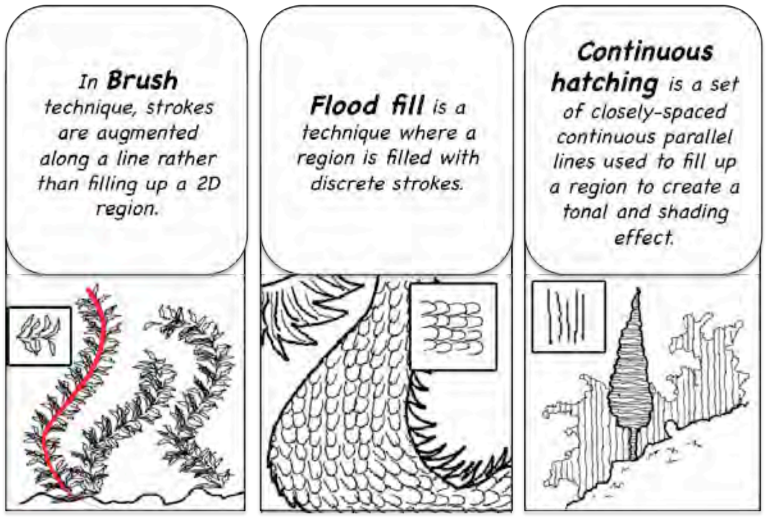
SandCanvas' digital features go beyond physical sand animation, such as recording, undo/redo, changing backlight, and mixed media. **Balance** is achieved by end-user programming, where users can **record a gesture** and **play the gesture** back by instantiating the gesture in different parts of the canvas. These digital capabilities enable faster and more dramatic transitions to create entirely new storytelling experiences.
Both professional artists and novices took part in **evaluation;** casual users liked the **instant gratification** of the medium and UI, while the experienced users effectively used the advanced features (e.g., mixed media, gesture playback) for compelling storytelling techniques that go **beyond** traditional sand art
The ability to use different parts of the hands to create visuals triggers **creativity** and **flow**
FLOW
Producing this new artistic medium required us develop a new approach to **real-time sand simulation** that strikes a balance between speed and realism. It also required us to design a simple and intuitive UI. When we analyzed it with respect to common sand animation techniques and our own taxonomy of gestures, we found it to be a genuinely rich artistic medium that enhances both **professionals'** and **novices'** opportunities for creative expression.
SandCanvas is available as an iPad app so that more creative hands can play with it.
SandCanvas was developed in NUS HCI lab with Chris Chua (NUS), Shengdong Zhao (NUS), Richard Davis (SMU), and Low-Kok LIM (NUS). Achieved Honorable mention and ACM Golden Mouse Award in **CHI 2011** and also appeared in **SIGGRAPH Talks 2011**
www.SandCanvas.com

1.3 Vignette

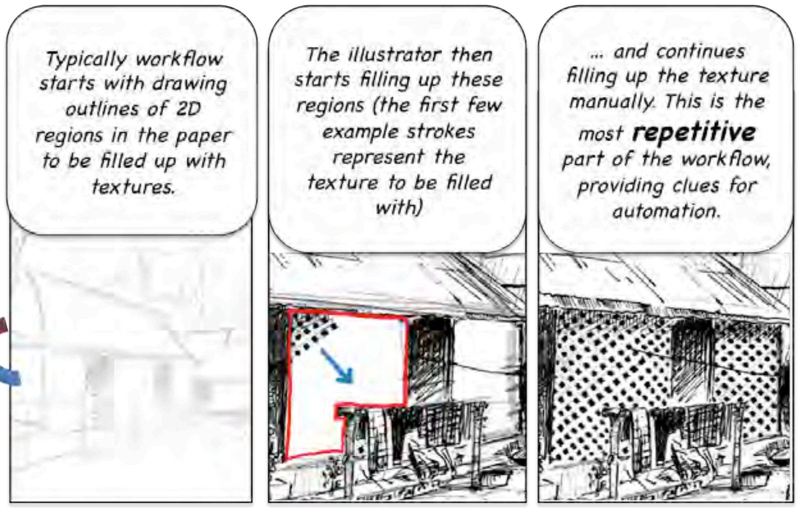


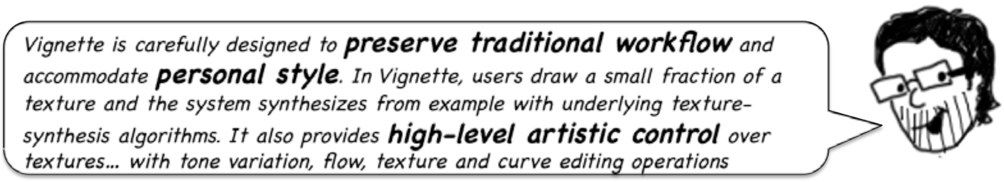
VIGNETTE ...
... is a style-preserving sketching tool for pen-and-ink illustration, with built-in texture synthesis capabilities.
Pen-and-ink illustration is a very popular art medium, incorporating a wealth of artistic styles and textures.
However, manually rendering pen-and-ink illustration requires inordinate amount of time, patience, and skill.
There are digital tools for texture illustrations, but they destroy the workflow, and often editing texture boils down to numeric parameter tweaking.
Furthermore, textures created with digital tools have a stiff mechanical look, in contrast to organic and stylistic rendering of original pen-and-ink illustrations.
To inform our design space, let's take a closer look at the medium, artifacts, and the process of texture-filling techniques in traditional pen-and-ink illustration
-- MEDIUM --
One of the key properties of pen illustration is the variety of artists' personal style... as Arthur Gutpill says, ...
"...The more conventional the art, the greater the opportunities for originality... pen drawing is akin to handwriting, and just as no two people write alike, so no two people draw alike."

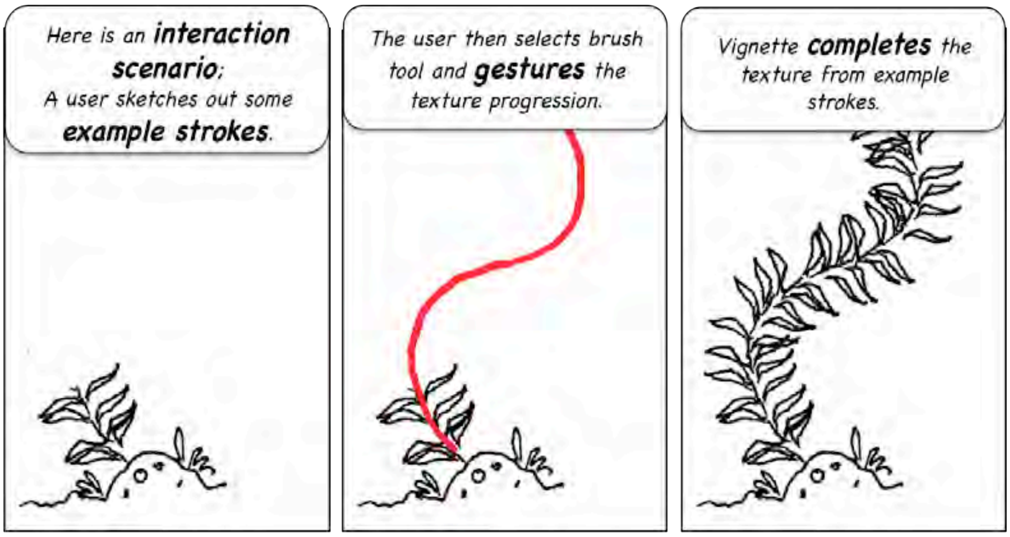
To understand the **ARTIFACTS**, we analyzed 56 illustrations by 32 artist and identified three major texture drawing techniques with room for automation.

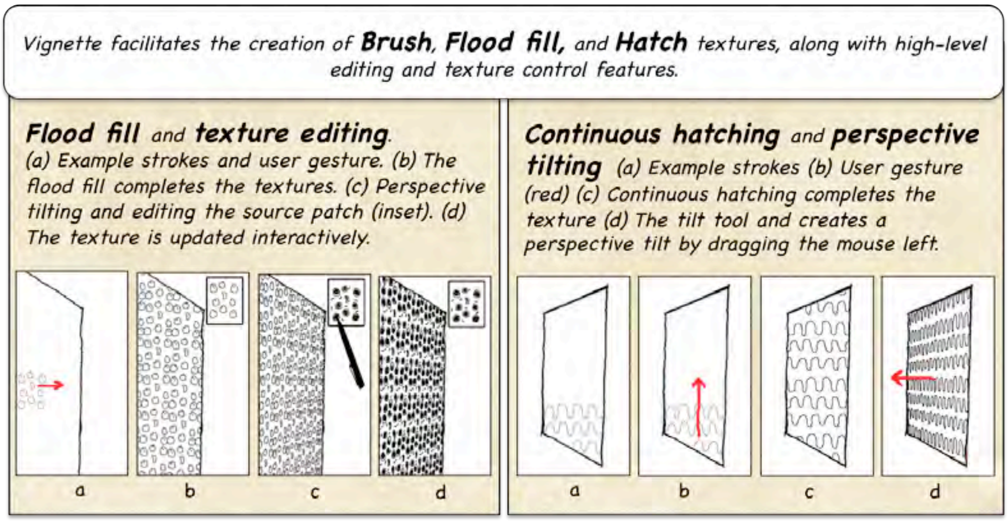


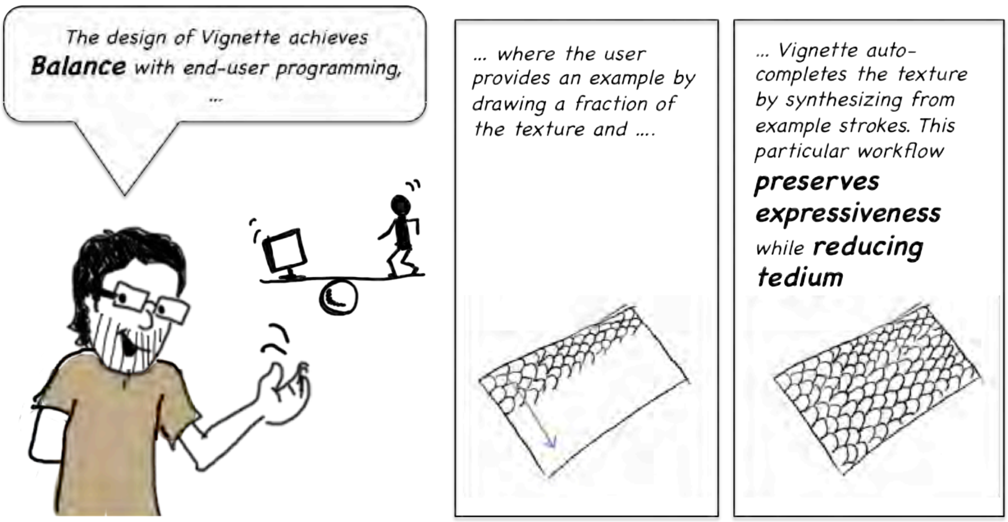
Let's now take a look at the **PROCESS** or **traditional workflow** for creating these textures.



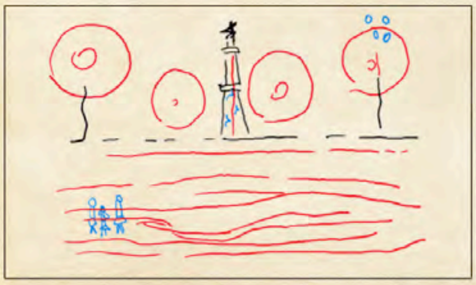


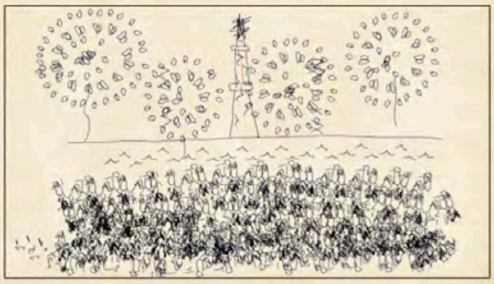

Here is an interaction scenario; A user sketches out some example strokes.
The user then selects brush tool and gestures the texture progression.
Vignette completes the texture from example strokes.


Vignette facilitates the creation of Brush, Flood fill, and Hatch textures, along with high-level editing and texture control features.
Flood fill and texture editing. (a) Example strokes and user gesture. (b) The flood fill completes the textures. (c) Perspective tilting and editing the source patch (inset). (d) The texture is updated interactively.
a
b
c
d
Continuous hatching and perspective tilting (a) Example strokes (b) User gesture (red) (c) Continuous hatching completes the texture (d) The tilt tool and creates a perspective tilt by dragging the mouse left.
a
b
c
d


The design of Vignette achieves Balance with end-user programming, ...
... where the user provides an example by drawing a fraction of the texture and ....
... Vignette auto-completes the texture by synthesizing from example strokes. This particular workflow preserves expressiveness while reducing tedium

Incorporating traditional workflow enables users to enjoy the **process** of creating texture and immerse themselves into drawing. A few scratches in the canvas can create textures in minutes.
The example strokes (blue) and gestures (red) drawn by a participant to produce an illustration in **5 minutes** (left). The final illustration (right)





While the ease of use makes Vignette exciting for novices, the artistic controls enable professionals to create high-quality and rich illustrations.

The example strokes (blue) and gestures (red) drawn by a participant to produce an illustration in **13 minutes** completely from scratch (top). The final illustration (bottom)

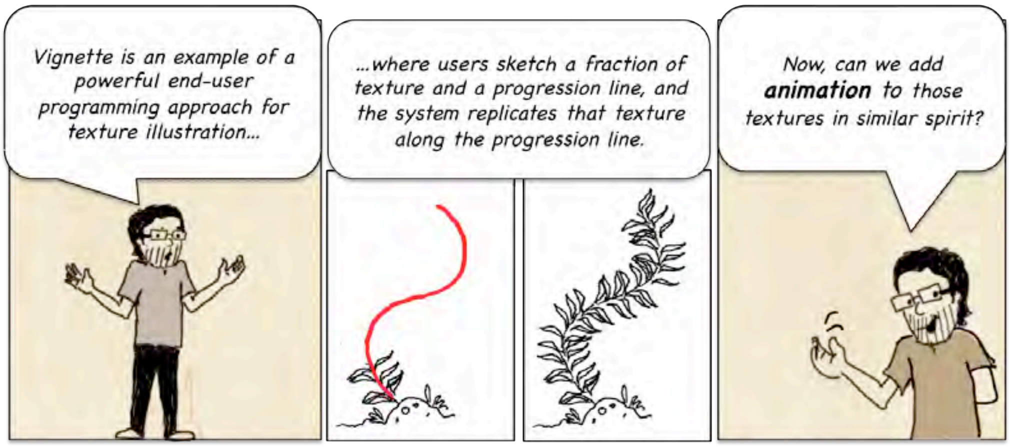
This drawing is done by a first-time user... took **19 minutes** to complete from scratch... look how organic it is! Preserving the personal style and sketchy rendering.

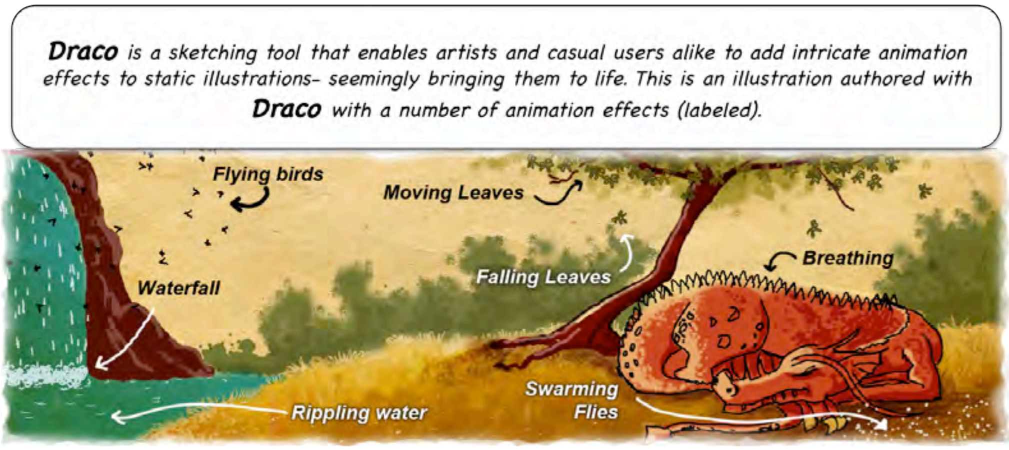
Illustration with Vignette in **24 mins**

Vignette was developed when I was visiting Japan Science And Technology Agency under Takeo Igarashi in Summer 2011. Collaborators: Shengdong Zhao (NUS) and Richard Davis (SMU). It appeared in **CHI 2012** and **SIGGRAPH Talks 2012**

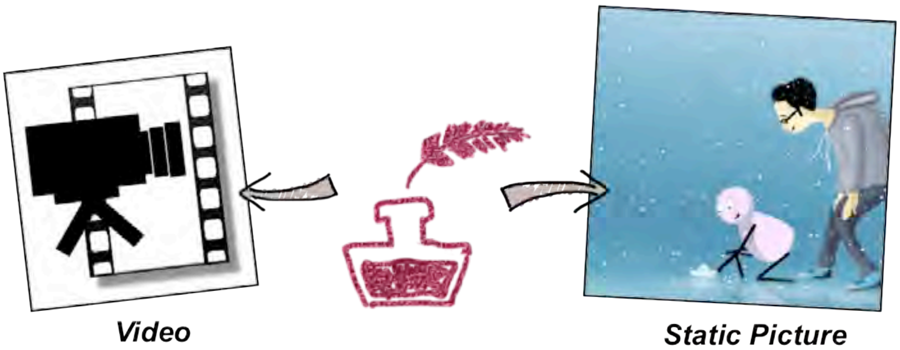
sketchvignette.wordpress.com/download

1.4 Draco





*As a **MEDIUM**, Draco takes inspiration from Cinemagraphs, fitting somewhere in between videos and static pictures. Illustrations with Draco capture the living qualities of a moment with continuous dynamic phenomena, yet exhibit the unique, timeless nature of a still picture.*

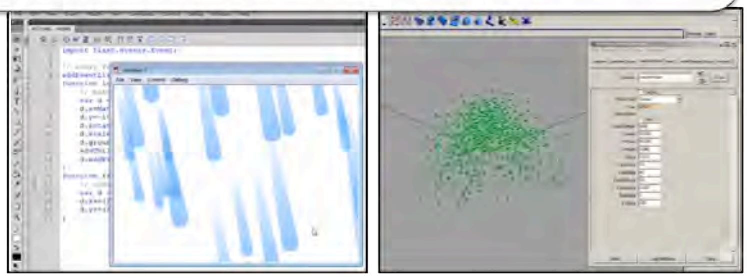


ARTIFACTS...

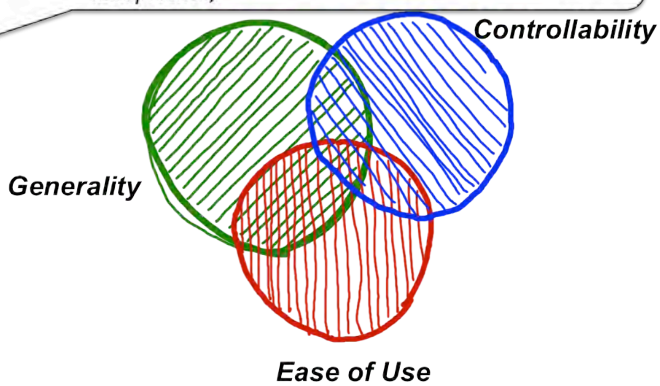
In this project, for analyzing ***Artifacts****, we limit our focus to groups of objects with coordinated motions. A variety of phenomena around us can be characterized as groups of objects with coordinated motion, such as snowflakes, tree leaves falling, water fountain, school of fish, and fan blowing off air.*

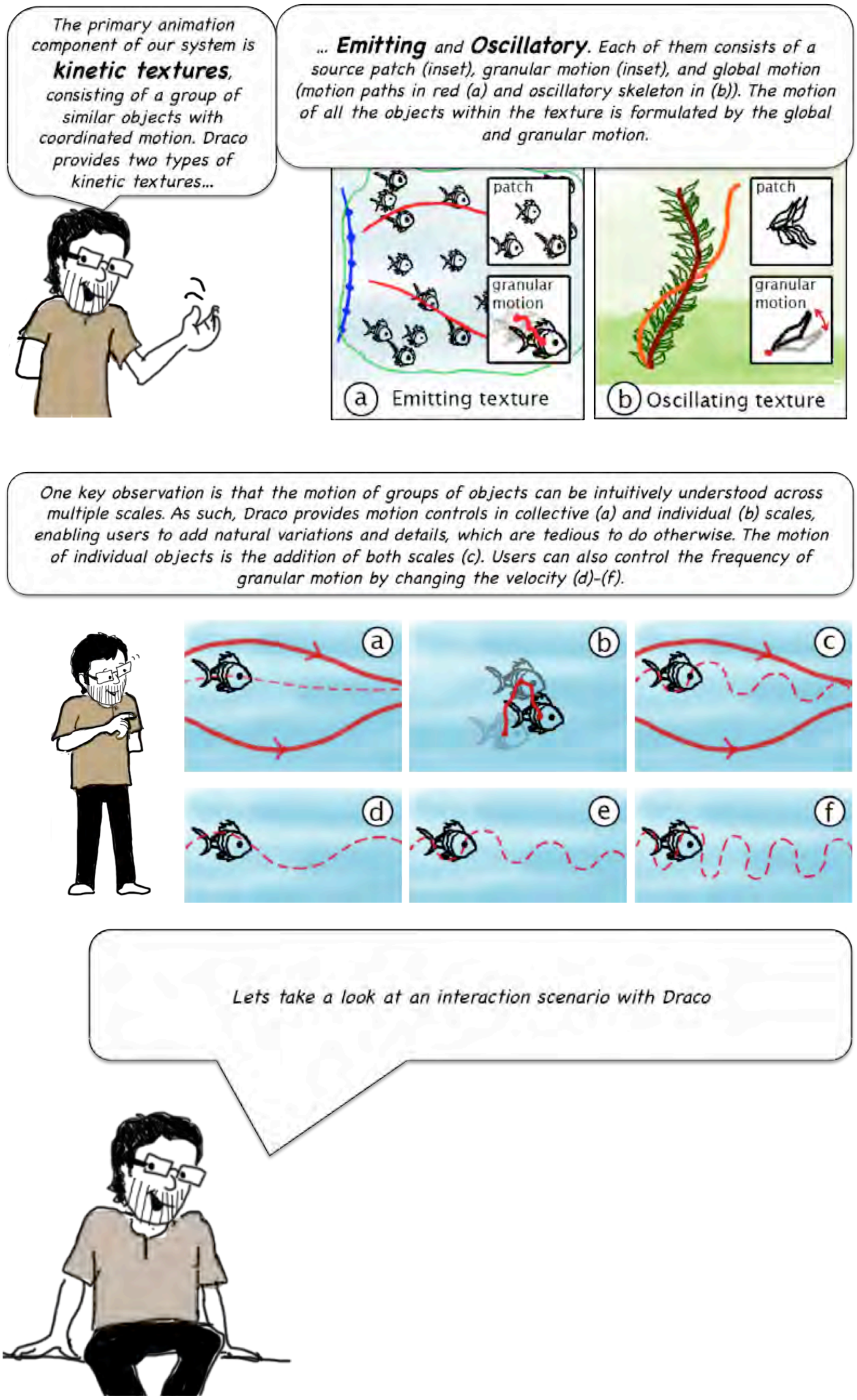
Creating such animation effects traditionally requires complex computer animation tools and creation ***PROCESS***

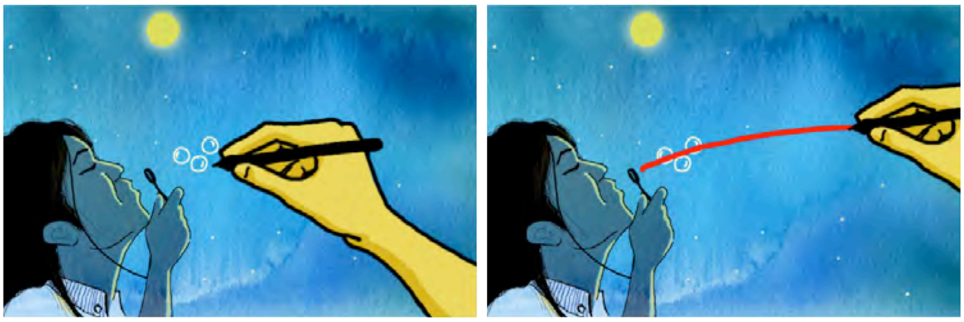
These tools are highly specialized and geared towards physical accuracy for professional animators. Furthermore, defining and controlling these behaviors typically require indirect controls, including programming scripts (left) and numerous parameters tweaking (right), which makes it difficult to rapidly prototype and experiment with motion effects.



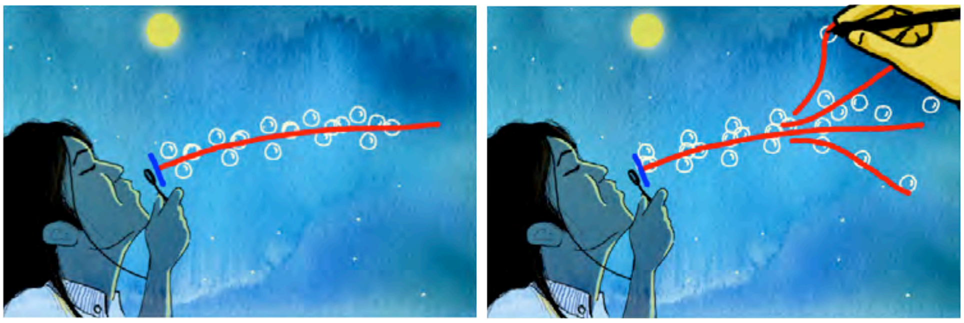
The goal of Draco was to provide an interface that simultaneously achieves generality (ability to create a wide range of effects, both natural and artistic), controllability (giving enough control to add details), and ease of use (leveraging the freeform nature of sketching and direct manipulation)



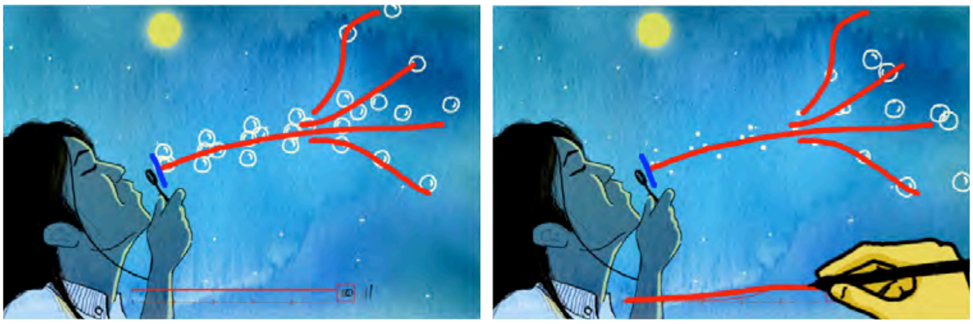

The primary animation component of our system is **kinetic textures**, consisting of a group of similar objects with coordinated motion. Draco provides two types of kinetic textures...
... **Emitting** and **Oscillatory**. Each of them consists of a source patch (inset), granular motion (inset), and global motion (motion paths in red (a) and oscillatory skeleton in (b)). The motion of all the objects within the texture is formulated by the global and granular motion.
patch
granular motion
a Emitting texture
patch
granular motion
b Oscillating texture
One key observation is that the motion of groups of objects can be intuitively understood across multiple scales. As such, Draco provides motion controls in collective (a) and individual (b) scales, enabling users to add natural variations and details, which are tedious to do otherwise. The motion of individual objects is the addition of both scales (c). Users can also control the frequency of granular motion by changing the velocity (d)-(f).
a
b
c
d
e
f
Lets take a look at an interaction scenario with Draco



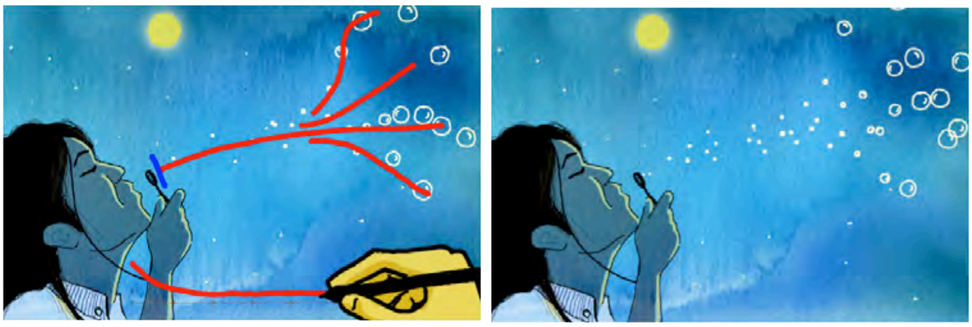
The user sketches example objects (left) and sketches the motion path (right)



The system generates an emitting texture with an emitter emitting bubbles moving towards the motion path (left). The user sketches further motion paths to spread the bubbles (right).

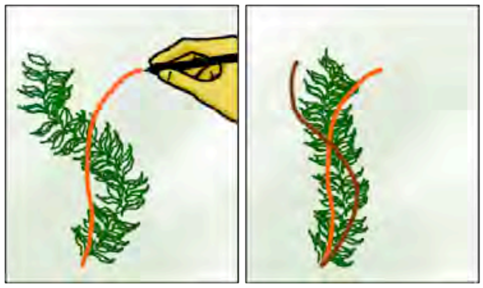


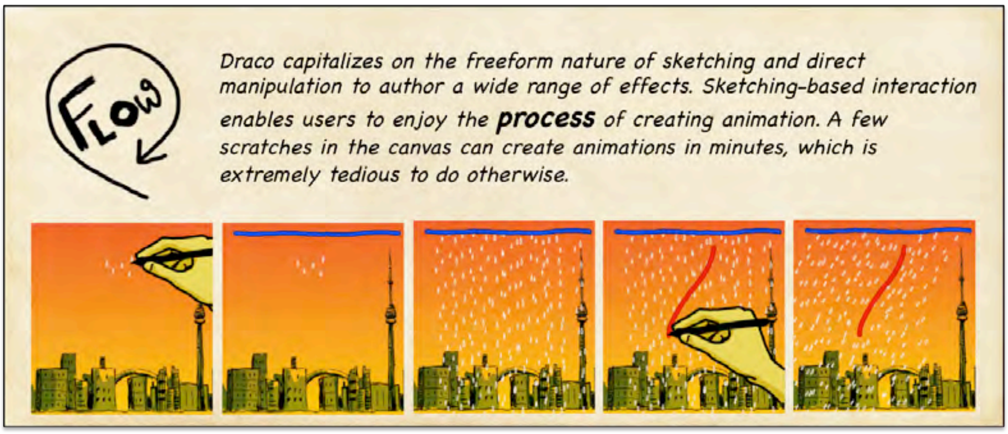
Finally, the user activates the motion profile widget (left) and sketches a new motion profile to adjust the scale of the bubbles along their path (right).

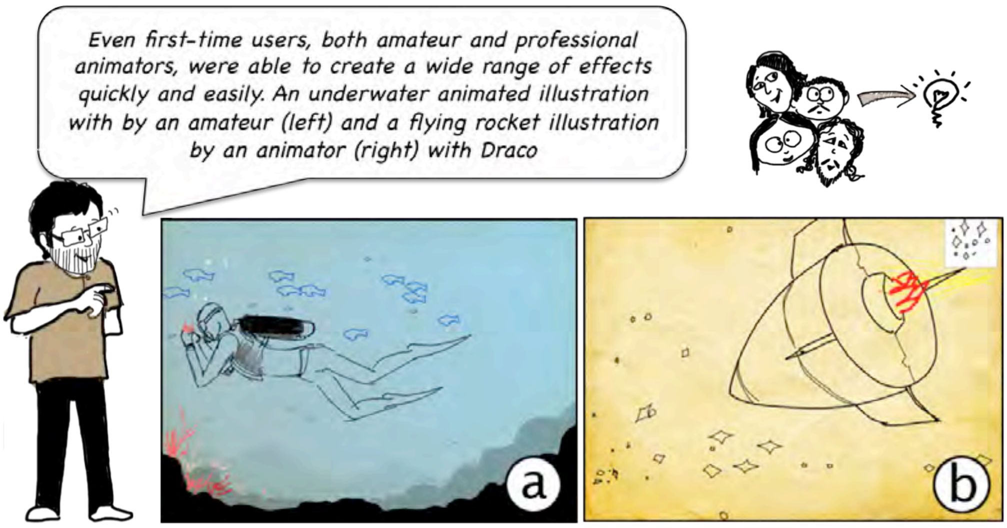


New velocity profiles make the bubbles grow and decelerate as they move away from the emitter (left). Final result (right).

To author a wide range of effects, Draco employs no underlying physics or simulation model. This ensures expressiveness and generality.

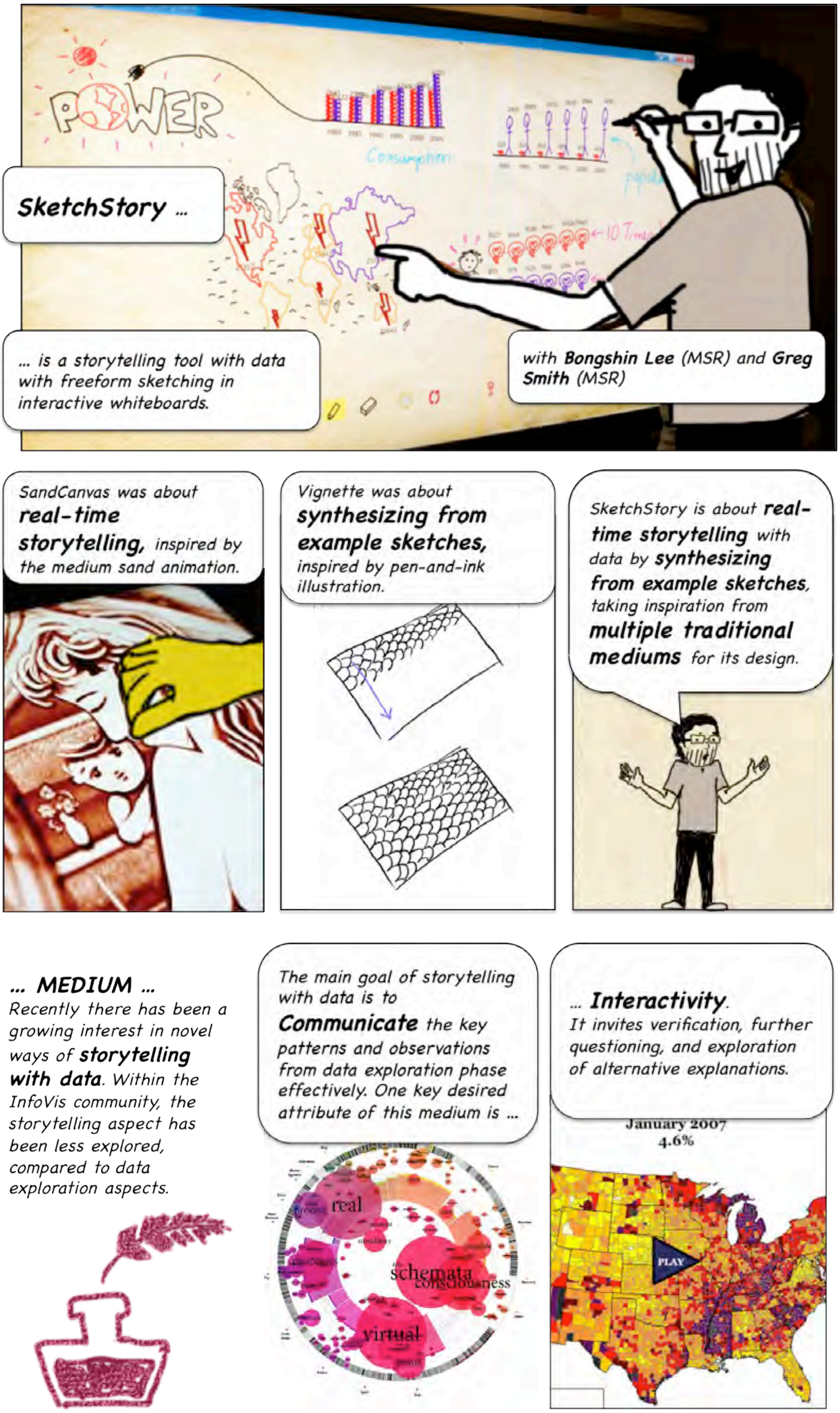






Draco was developed in Autodesk Research with Fanny Chevalier, Tovi Grossman, Shengdong Zhao and George Fitzmaurice. It won three awards (Best paper honorable mention award, Best Talk award and Golden Mouse award) in **CHI 2014.**
www.autodeskresearch.com/projects/draco

1.5 SketchStory



For the ***PROCESS,*** *SketchStory takes inspiration from* ***whiteboard animation****, where presenter sketches along with synchronized narration to vividly tell a story.*

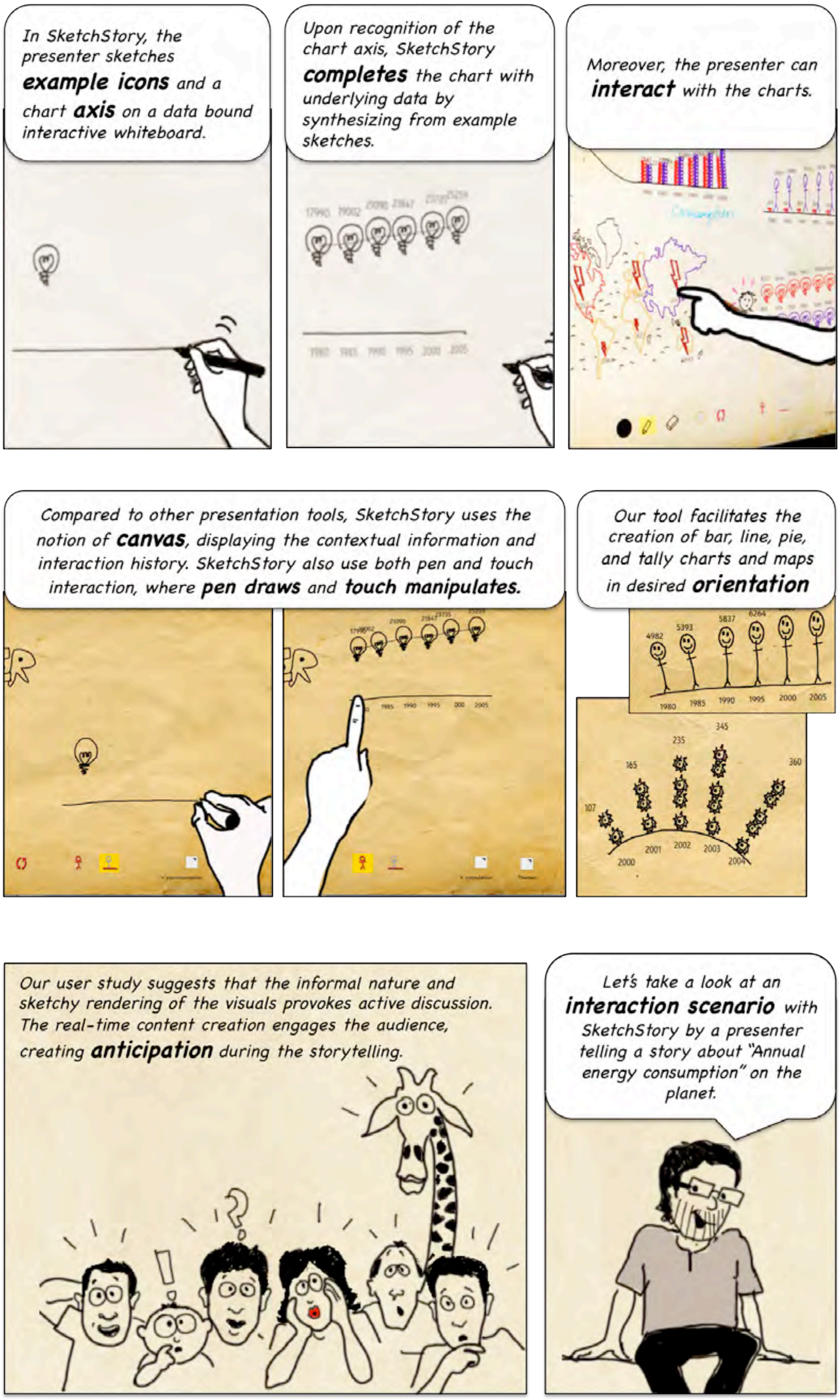
The expressive graphic style makes whiteboard animation a very unique and ***engaging*** *form of storytelling. It has become increasingly popular in domains such as advertising and education.*

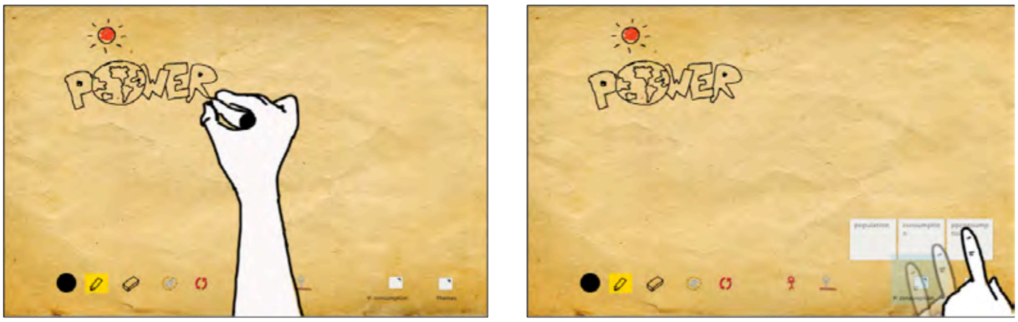
Real-time *dynamic* ***sketching*** *conveys the order of action sequences by directing viewer attention from one object to the next, building anticipation as in a story - making it an effective storytelling technique to plan ideas, emotions, and thoughts.*

The inspiration for the ***ARTIFACTS*** *comes from* ***Infographics****- graphical representations of information, data, or knowledge.*

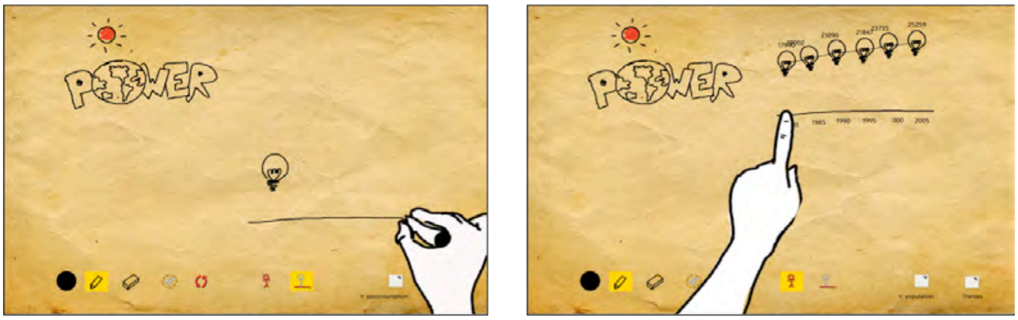
Infographics communicate complex information in an aesthetically pleasing way, employing icons and other visual elements that are customized to the dataset ... Infographics allow presenters the artistic freedom to create a unique, ***personalized chart,*** *taking full advantage of an innate visual language that is largely universal.*

The goal was to create a new, more engaging way of telling stories with data by inheriting and ***melding*** *the desirable properties from Data Visualization, Whiteboard Animation and Infographics.*

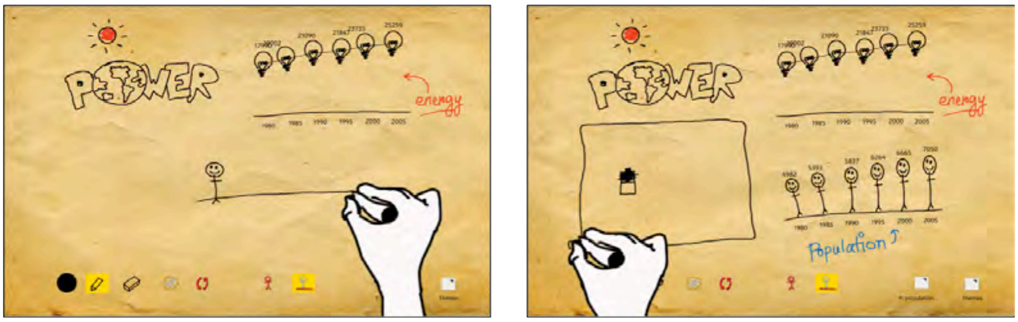

In SketchStory, the presenter sketches **example icons** and a chart **axis** on a data bound interactive whiteboard.
Upon recognition of the chart axis, SketchStory **completes** the chart with underlying data by synthesizing from example sketches.
Moreover, the presenter can **interact** with the charts.
Compared to other presentation tools, SketchStory uses the notion of **canvas**, displaying the contextual information and interaction history. SketchStory also use both pen and touch interaction, where **pen draws** and **touch manipulates.**
Our tool facilitates the creation of bar, line, pie, and tally charts and maps in desired **orientation**
Our user study suggests that the informal nature and sketchy rendering of the visuals provokes active discussion. The real-time content creation engages the audience, creating **anticipation** during the storytelling.
Let's take a look at an **interaction scenario** with SketchStory by a presenter telling a story about "Annual energy consumption" on the planet.



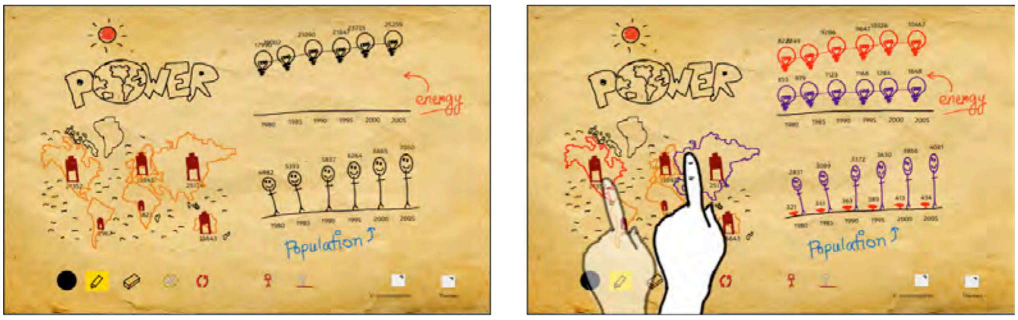
The presenter sketching the title of the story (left) and selecting a data attribute, "individual energy consumption" (right).



Sketching an example icon and axis (left). SketchStory completes the chart from the example sketch. The presenter moves the line chart by touch (right).



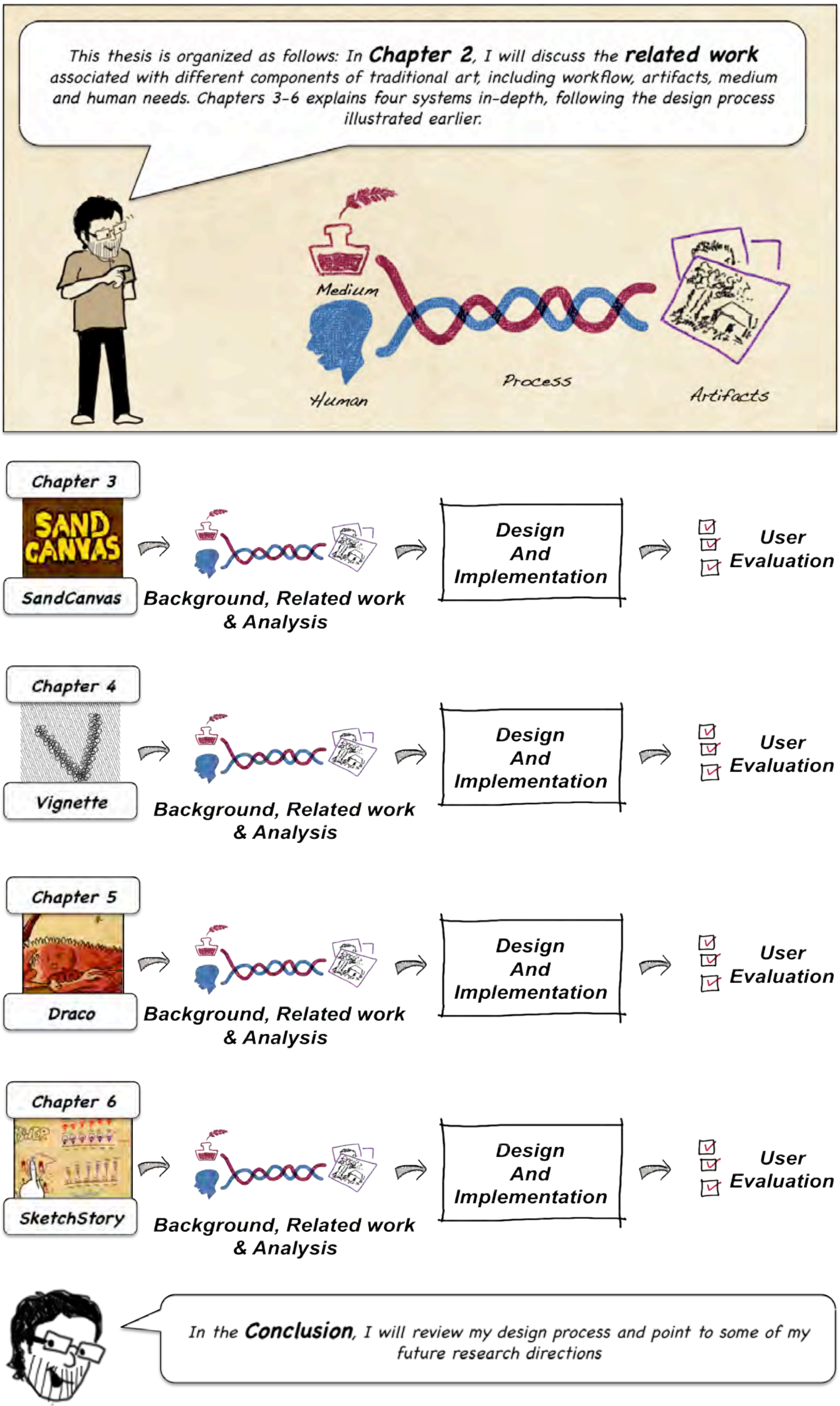
Sketching an example icon and axis for a data attribute, "population" (left). Automatically generated bar chart for the population. Sketching an example icon and axis for a data attribute "energy consumption" (right).



Automatically generated map with overlaid consumption information across the continents (left). Selecting America and Asia for comparison updates other charts (right).

> ***SketchStory*** *was developed in MSR (Redmond) with Bongshin Lee and Greg Smith. It appeared in InfoVis 2013, TechFest 2013, and received a lot of media coverage.*

1.6 Organization



2. RELATED WORK

This thesis presents four computational art and communication tools that facilitate creative media authoring and novel storytelling techniques: SandCanvas in Chapter 3, Vignette in Chapter 4, Draco in Chapter 5, and SketchStory in Chapter 6. When designing interactions in these tools, I sought inspiration and insight from traditional craft (e.g., sand animation, ink illustrations) and existing media (e.g., cinemagraph [24], whiteboard animation, infographics). The projects in this thesis were also inspired by numerous new media interactive installations and advances in computer graphic and human-computer interaction techniques.

In this chapter, I will start by discussing the different aspects of traditional art media, such as raw materials, tools, and techniques. I will then illustrate key affordances and characteristics of traditional art media and argue to embrace those affordances in digital art media. In Section 2.2, I will then portray initial research and future directions on scientific understanding of art in different disciplines - including neuroscience, art literature, psychology, and computer science. Section 2.3 briefly summarizes representative works in the field of human-computer interaction for creative media authoring, including how direct manipulation user interfaces, tangible user interfaces, and sketch-based user interfaces facilitate digital content creation across a range of creativity support tasks. In Section 2.4, I will discuss representative works in the computer graphics community for artistic rendering, commonly referred as non-photorealistic rendering (NPR). I will discuss the historical perspective of NPR and how it has been widely adopted in a broad range of domains, including artistic tools, movie production, game production, presentation, visualization and beyond. Much of the research done in this domain heavily emphasizes the rendering and final visual output, compared to the creative process of creating those artifacts. Finally, I will conclude by emphasizing a holistic understanding that provides significant insight into designing new media digital art tools, justifying my approach that considers process, artifacts, and media altogether. Related works that are more specific to a particular project are discussed in the corresponding chapter (for example, related work in sand rendering is specific to project SandCanvas, hence it is discussed in Chapter 3).

2.1 Traditional Art Media

In general, the definition of art can be very loose. Anything creative or expressive can be considered as an art. Any material or tool in the hand of an artist can result in artwork. In this section, by "art," I am referring to 2D drawings, paintings, graphic design, and illustration.

The history of traditional art media dates back hundreds of years. Our ancestors built sophisticated tools, and created skills and techniques to transform the materials of nature into art.

While not exhaustive, this section briefly describes different aspects (e.g., raw material, painting tools, and crafting techniques) of some of the most popular and representative 2D art mediums, such as oil painting, watercolor, pencil sketching, ink illustration, and sand animation. The unique combination of different materials, tools, and techniques defines a unique art medium.

2.1.1 Raw Materials of Art

The importance of understanding the nature of the materials one uses in painting cannot be underestimated. Without some basic knowledge of the working properties of materials, artists may not fully develop and progress with their work.
– Pep Seymour [143]

Oil painting, watercolor painting, acrylic painting, and soft pastel painting are some of the most prominent and popular means of traditional art media. The fundamental drawing element of these media is pigment, a material that is composed of finely divided particle structure and changes the color of reflected or transmitted light as the result of wavelength-selective absorption. Consequently, pigments are easier to control when mixed with binding media to create stable pastes of color. This color is applied by brushing or spreading onto a surface. Pigments usually have a greater resistance to fading on exposure to sunlight. There are a number of factors associated with a pigment that can make it suitable for artistic applications, such as regularity in particle sizes, free of impurity, free of additives, response to light, and permanence to atmospheric conditions. Understanding and choosing the right pigments is crucial to anyone learning to paint or involved in the practice of painting [143]. For each medium (oil painting, acrylic, soft paste, watercolor), the characteristics of chosen pigment are different from each other. Different pigments use different wetting agents to aid dispersion in the chosen paint binder. For instance, oil painting pigments use mineral spirits to disperse the particles into drying oil, such linseed oil, poppy seed oil, walnut oil, safflower oil, etc, while acrylic color pigments use special synthetic wetting agents [143].

Pen-and-ink illustration, yet another very popular form of art medium, uses ink as a raw material. In general terms, any drawing medium with liquid form can be termed as ink. However, ink illustration generally uses the simplest form of ink, generally used more for writing and made from gallnuts. Often, fine particle pigments are mixed with a thick solution of gum arabic to create hand-made colored inks, which are also used by pen illustrators.

In sand animation, a special type of fine-grained and rinsed sand is used as the primary material. Some sand artists also use colored sand/powder for more expressive textures. Sand artists also use a special sandbox setup with backlit box, rinsed sand, brushes, spatula, etc.

The abundance of choices of materials, tools, and painting techniques make traditional art medium so expressive that each artist can grow and develop their own distinct personal style. Understanding how the material properties are attributed to the chosen medium is the starting point for any artist or designer. For instance, in sand animation, artists use fine-grained sands, instead of coarse-grained sand or any other materials, mainly for two reasons: Firstly, due to its fine nature, it creates a texture gradient with the underlying light-box, which would be difficult to produce otherwise. Secondly, fine-grained sand bears unique dynamics, which makes sand animation enjoyable and pleasant to watch. Typically, computer graphics researchers intend to apply physics-based simulation models to simulate artistic rendering in the digital domain. There is a number of sand simulation models based on particle systems and Newtonian physics. But, due to its fine nature, the number of particles within a given canvas was huge, which makes it challenging to make it real-time. As such, in SandCanvas, we used an empirical approach that strikes a balance between performance and similarity to fine-grained sand. For a more comprehensive discussion in artistic rendering, see Section 2.3 (Research in Stylized Graphical Rendering).

2.1.2 Tools and Techniques

> *The more conventional the art, the greater the opportunities for originality. We might go so far as to say that there is perhaps no medium offering one a better chance for development of a personal technique than the pen, for pen drawing is akin to handwriting, and just as no two people write alike, so no two people draw alike…* Arthur L. Guptill [48].

Tools

Drawing tools are used in performing an operation in the practice of drawing or painting. The choice of drawing tools today is vast, as are the types of support applicable to drawing.

For oil painting, watercolor, and acrylic painting, the most common form of tool is the paintbrush. A paintbrush is a brush for applying ink or paint. These brushes are usually made by clamping the bristles to a handle with a ferrule. Brushes may come in many shapes (e.g, round, flat, bright, filbert, fan, angle, mop, rigger, stippler, liner, dagger, scripts, Egbert) and sizes (**Figure 2-1**). The bristles of the brush may vary according to the medium material. For instance, watercolor bristles are made of sable or nylon, oil painting bristles are made of sable, and acrylic brushes are almost entirely synthetic.

Figure 2-1: Different types of paintbrushes

Even a narrow craft like pen-and-ink illustration has a number of variables – thickness, precision, boldness, character, and emotion - providing artists a vast zone to develop their own distinctive artistic style [101]. In pen-and-ink illustrations, artists typically use a wide range of pens, including calligraphy and lettering pens, drawing pens, gel ink pens, marking pens, technical pens, and writing pens. The nature of lines varies with different types of pens. The choices of pens are highly subjected to artistic styles and personal preference.

In sand animation, sand animators use hands and sand only to generate a wide range of shapes and drawings. In this case, the artist's hand acts as a dynamic brush, using fingertip, finger carve, palm, and other parts of the hand to generate different types and shapes of visual effects. Chapter 3 presents an elaborate discussion on the different types of gestures used by sand animators.

In addition to these tools, artists often use dry materials (such as pencil or charcoal) and other auxiliary prosthetics (spatula, plate) for rough layout and other purposes.

Mainstream professional digital art tools, such as Photoshop, Illustrator, and Maya, are primary designed for WIMP-based interfaces, where the abundance of choices over materials, tools, and techniques are translated into numerous parameters and widgets. In traditional art process, artists can capitalize on the direct manipulation and physical affordances. In contrast, for most of the digital art media, the input bandwidth is limited to mouse, stylus, and keyboards. For instance, Sketchbook Pro consists of a tool palette, providing all different types of pencils, pens, brushes, markers, and custom stamps. Each of these tools consist of a parameter widget to adjust the opacity and size. A user has to explicitly switch into different modes to change the types of tools or brushes. While these digital tools give artists greater freedom in terms of choices, the limited input bandwidth to explore the vast space of parameters disrupts the drawing experience. As such, the limited input bandwidth of the digital art tools lose some of the "physical affordances" of traditional art tools. However, recent advances in sensing technologies, such as touché [137], context sensing with pen computing [148], depth sensing, pressure sending, stylus motion sensing with stylus [54], and haptic feedback capabilities [12], open up new possibilities for digital art designers to capture artists' intent more seamlessly and capitalize on the physical affordances. With context sending, for instance, Song et al. [148] explored a multi-touch stylus to capture the user's gestures within the pen. This multi-touch sensor enables new touch gestures to be performed and detects how the users grip the device as a mechanism for mode switching. In this case, the sensor can detect different types of distinguished grip, such as relaxed grip, tripod grip, tuck grip, sketch grip, and wrap grip, and switch to the corresponding mode accordingly. Another example of increased bandwidth for artistic expression is SandCanvas (Chapter 3). Unlike traditional capacitive touch screens, the SandCanvas user interface uses a higher input bandwidth with computer vision techniques to capture the shape and size of human hand touch regions. This enables artists to directly manipulate the canvas and capitalize on the expressive vocabulary of gestures at their disposal without explicitly changing modes, creating different types of effects.

Techniques

> *In relation to the activity of painting, the application of technique can be a crucial factor in the complete realization of painted images.* – Pep Seymour [143]

Drawing techniques are specialized procedures and methods to achieve desired artistic effects. Even if artists pick the exact same medium, materials, and tools, their technique may vary significantly. For instance, during the Renaissance, there were four canonical painting modes for oil painting - *Unione*, *Sfumato*, *Chiaroscuro* and *Cangiante*. The most prominent practitioner of Sfumato was Leonardo da Vinci, and his famous painting of the *Mona Lisa* exhibits this technique. Most oil painters paint in layers that are known as "glazes." The method was first perfected through an adaptation of the egg tempera painting technique and was applied by the Flemish painters in Northern Europe with pigments ground in linseed oil. More recently, this approach has been called the "Mixed Technique." The first coat (also called "underpainting") is laid down, often painted with egg tempera or turpentine-thinned paint. This layer helps to "tone" the canvas and to cover the white of the gesso. Many artists use this layer to sketch out the composition. Artists in later periods often used this wet-on-wet method more widely, blending the wet paint on the canvas without following the Renaissance-era approach of layering and glazing.

In the context of pen-and-ink illustrations, strokes and textures are the building blocks of drawings. However, different artists use different techniques when utilizing these building blocks for pen-and-ink illustration. Just like strokes, textures and tones are extremely expressive, and master artists often develop their own styles and techniques that distinguish their works from others [10]. For example, **Figure 2-2** (left) is composed of many lines; the technique is restrained and controlled by the well-distributed tone values. This type of technique is called tight drawing technique [48], where drawings' shapes are precise and specifically placed. On the other end of the spectrum, for instance, the illustration in **Figure 2-2** (right) is largely expressed by shadow tones. Many edges of the planes are only suggested, much being left to the imagination. Hence, the drawings are economical in line and tone. This type of technique is referred to as loose drawing technique [93]. Chapter 4 presents an elaborate discussion about the different types of texture used in pen-and-ink illustrations.

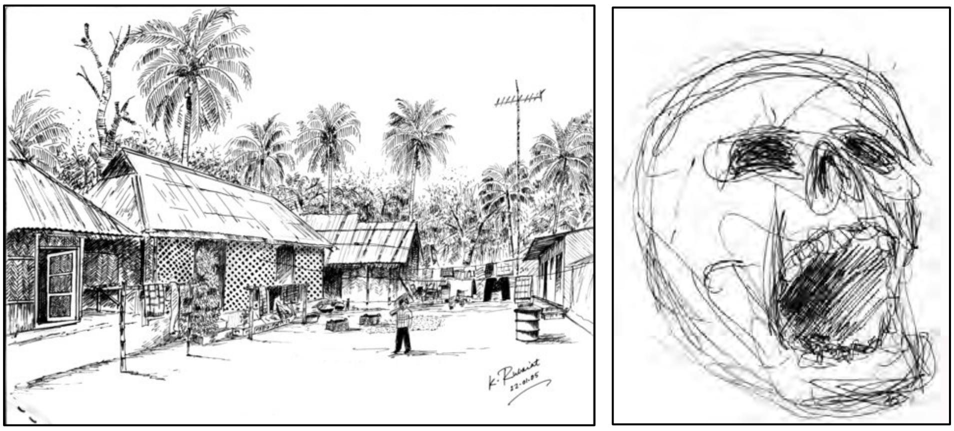


Figure 2-2: Tight (left) and loose (right) drawing techniques in pen illustration

As for techniques, visual artifacts can vary significantly due to difference in artistic preferences and styles, even within the same medium. In this regard, technique is highly adhered to an artist's input and drawing process. In the computer graphics community, a number of artistic style rendering techniques are completely automated or solely based on parameter controls, overlooking the value of individual styles, preferences, and techniques. The design process of the tools within this thesis, thus, capitalizes on the powerful notion of end-user programming, where the user sketches the sample, and the system reduces tedium by synthesizing from example sketches with gesture stroke (Vignette, Draco) and underlying data (SketchStory). Apart from the notion of style preservation, users tend to have preference over the types of tools used to create the final artifacts. The creation of artifacts is not automated, users have the freedom the employ their personal style and individual drawing techniques. Our user study with SandCanvas (Chapter 3) suggests that users tend to have a personal gesture profile, where individuals use certain gestures more frequently than others. We also observed two approaches to creating artworks in SandCanvas, each with a different distribution of drawing techniques. In the free task, five users took a subtractive approach, in which sand is manipulated to create shapes. Two users took an additive approach, in which shapes are made by pouring sand onto the canvas. In our user study with Vignette (Chapter 4), one artist made heavy use of hatching effects, while the other three used brush technique for the illustrations.

2.2 The Science of Understanding Art

How does art work?

How does art rendering express emotion, mood, motion, and form?

What is the process of creating artistic imagery?

Despite centuries of interest, these questions remain unanswered. These questions have been raised in many disciplines, including psychology, cognitive science, neuroscience, and more recently, in human-computer interaction and engineering. As a human-computer interaction researcher and designer, my primary interest is in designing and developing computational tools for creative media authoring. Yet, understanding these questions can yield significant insights into art, illustration, creative process, and lead us to design new kinds of art.

In spite of the fact that these questions are long-term questions and can take years (even decades) to address, there is excellent initial research building the foundation to the quest. A number of scientific studies, analysis, and research across different disciplines intend to understand how art works and how viewers respond to artistic images.

2.2.1 Neuro-aesthetics: Study of art by neuroscientists

Although we rarely confuse a painting for the scene it presents, we are often taken in by the vividness of the lighting and the three dimensional (3D) layout it captures. This is not surprising for a photorealistic painting, but even very abstract paintings can convey a striking sense of space and light, despite remarkable deviations from realism. – Patrick Cavanagh [20]

There have been a number of efforts to understand art by neuroscientists. Cavanagh argued that artists employ an ”alternate physics” by deviating from the physical world in their drawings [20]. The impossible shadows, shapes, colors, and contours do not interfere with the viewer's understanding of the 3D scene. This reveals that our brain uses a simple, reduced physics to understand the world.

> *The purpose of art, surely, is not merely to depict or represent reality—for that can be accomplished very easily with a camera—but to enhance, transcend, or indeed even to distort reality. . . . What the artist tries to do (either consciously or unconsciously) is to not only capture the essence of something but also to amplify it in order to more powerfully activate the same neural mechanisms that would be activated by the original object* – Ramachandra [188]

Art, in this regard, is a type of found science where artists tap into shortcuts to amplify our neural mechanisms. Artists typically use a combination of visual elements – line, light and shadow, shapes, depth, time and motion, color, etc.

Lines and Line drawing

A few lines of drawing can convey shape, emotion, depth, and expression. Line drawings are extremely simple but remarkably effective. The effectiveness of line drawing is not simply attributed to learned convention or passed through culture. How does line drawing work? How does it communicate to the brain? Studies show that conventional line drawings do not just trace out the brightness discontinuities or contour or shadow pigments in an image, rather, artists identify the key contours to be perceived by the human brain to understand the structure of the object. Recent studies by Cole et al. [26] on how we view drawings and where do people draw lines points to the fact that the goal of line drawing is to produce a minimal set of lines so that the desired 3D shape is reconstructed by the viewer.

Shadow and lighting

Even though shadows in the real world are subjected to many constraints, artists take many liberties while depicting shadows without disrupting the original light appearance, space, and form of the scene. Perceptual experiments [20] suggest that artists use darker shadows than their immediate surroundings. Studies also suggest that inconsistent lighting direction and color is not readily noticed by the viewers [81].

Representing Time and Motion

Artists and scientists use a number of visual techniques to depict motion and time in static pictures. In fact, often the pedagogical purpose of the representation of motion in static form is more important than the motion itself. Cutting [182] illustrated five different types of distinct representations of motion - *dynamic balance*, *multiple images*, *affine shear*, *blur* and *vector like lines* superimposed on images. Each of these techniques has their own affordances and limitations. Cutting [182] discussed the efficacy of these techniques in terms of four criteria, increasingly focused on the necessities of science – *evocativeness*, *clarity* of object, *direction* of motion and *precision* of motion. However, Cutting argued that the neurophysiological underpinnings of the efficiency of motion representations are difficult to measure. Instead, these techniques merely suggest to an observer that a motion had occurred.

Shapes, Postures, and Depth:

Perceptual studies indicate that certain shapes, colors, and composition evoke different emotion and reaction in a viewer's mind. Skilled and experienced artists make use of these visual cues to express emotion, mood, and clarity. Character artists typically use shape, size, pose, color, and proportion as the first design layer to express role, physicality, and personality traits of a character [62]. For instance, circular shapes represent love, caring, and tenderness. In contrast, an inverted triangle shape represents strength, power, and stability. Even though there is no Golden Rule that quantifies a generic design template for stylized character figure drawing, the use of pose, form, shapes, and colors to represent character traits, emotion, and mood is well established in art and psychology literature [62, 14].

Perceptual studies can lead to insightful user interface design. For instance, perspective-drawing techniques are commonly used for conveying depth. Recent experiments by Schmidt et al. [138] indicate that some of the widespread assumptions about 3D illustrations are incorrect. Based on their study findings, they analyzed current 3D curve drawing techniques for susceptibility to foreshortening bias and made some suggestions for future sketch-based modeling systems. However, perceptual studies for user interface design insights and evaluating *task-oriented* and *artistic goals* remain as a future work that would help us to understand art even better.

2.2.2 Descriptive Art Analysis

Mainstream analysis of art, in particular art history, is highly *descriptive* [20, 155, 153, 99]. In general, most of the *descriptive* theories are qualitative. Art critics and practitioners discuss the history, general features, trends, artistic styles, affordance, and intentions of a particular art medium or a particular time. For example, the seminal book of Edward Tufte's *Envisioning Information* [155] is an excellent reference. It provides practical advice about how to explain a number of design strategies with examples to convey multi-dimensional, dynamic, and moving information by visual means, with extraordinary examples to illustrate the fundamental principles of information displays. It also provides insightful commentary over a number of prominent examples including maps, charts, scientific presentations, diagrams, computer interfaces, statistical graphics and tables, stereo photographs, guidebooks, courtroom exhibits, timetables, use of color, a pop-up, and many other historical displays of information. Similarly, Scott McCloud's *Understanding Comics: An Invisible Art* [99] explores the historical context, fundamental vocabulary, and formal aspects of comics and the various ways in which these elements have been used. This work explicates theoretical ideas about comics as a medium of art and communication. In a similar vein, Disney Animations *Illusion of Life* [153] illustrates the process and philosophy of Disney master animators with twelve basic sets of animation principles - including squash and stretch, anticipation, staging, straight ahead action and pose to pose, follow through and overlapping, slow in and out, arcs, exaggeration, appeal, secondary action, and timing. The *descriptive* analysis of art media has inspired historians, artists, designers, and computer scientists alike. Descriptive analyses, among many others, have a direct and profound impact on a number of research tools, algorithms, and user interface designs in the fields of computer graphics, computer animation, and information visualization. In spite of the fact that the *descriptive* mainstream art literature has a profound impact on computational research and tools, they lack a sufficiently precise and formal language for implementation that a computer scientist would seek. Hertzmann argued that non-photorealistic rendering (NPR) research would play a key role in the scientific understanding of visual art and illustration and provide that formal language to describe art [51].

2.2.3 Generative Art Theories

In contrast, *generative* theories describe how to create specific types of imagery [51]. *Generative* theories are mostly quantitative, providing a more formal and structured language (algorithms) and parameters. They do not apply as broadly to many different styles. A large number of *generative* theories are emanated from attempts to develop algorithms and artificial intelligence, enabling computers to generate art. Professor Harold Cohen, UCSD, developed one of the pioneering artificial intelligence/artist named AARON in the mid-1970s [25]. Cohen's initial question while developing AARON was: *What are the minimum conditions under which a set of marks functions as an image*? Initial versions of AARON created abstract drawings that grew more complex through the 1970s. More representational imagery (e.g, plants, rocks, humans) were added in the 1980s. AARON cannot learn new styles or imagery on its own. New capabilities required manual coding by Harold Cohen, but AARON is capable of producing a practically infinite supply of distinct images in its own style. The resulting outputs have been exhibited in galleries worldwide; the exhibitions serving as an artistic equivalent of the Turing test.

However, AARON's output follows a noticeable formula.

More recently, the NPR community has emphasized *generative* theories of art to produce artistic styles and compute viewers' response [25]. Broadly, NPR algorithms are aimed at two types of goals [51]: *task-oriented goals* and *artistic goals*.

NPR algorithms with *task-oriented goals* communicate specific information, such as shape, spatial location, and relationships. For such objectives, researchers can formulate an optimization by a set of objective terms that measure specific goals that are relevant to that task, including perception of shape or other properties of image.

NPR algorithms with *artistic goals* create images that are beautiful or expressive. Quantifying the *artistic goal* is a difficult problem, since aesthetic responses can vary from person to person and culture to culture. Instead, some researchers have assessed related measures, such as how image stylization affects eye gaze [36] and how image stylization algorithms affect memory or learning tasks [170]. Getting informal feedback from practitioners about aesthetic goals of an algorithm can be extremely valuable for developing ideas and techniques, but it would be misleading to consider it as rigorous evaluation criteria [51].

Descriptive theories and generative theories of art drive each other in various ways [46]. Descriptive theories hint about the overall factors to take into account, defining terms and evaluation criteria, while the trial-and-error of designing generative models can lead to insights for creating concrete language to describe and create artworks [51]. The projects discussed within this thesis relied on descriptive and generative art analysis for designing the new media art tools. For instance, while designing Vignette (Chapter 4), descriptive art books on pen-and-ink illustration aided the development of features and workflow. Additionally, previous generative algorithms for texture synthesis aided the development of new algorithms.

2.3 Research in Interaction Design

In recent years, HCI researchers have explored the design of digital art systems that bridge the gap between physical and digital art. From a design perspective, I seek to understand the traditional art process and the different attributes. I believe that these attributes should draw more attention from digital art tool designers and the NPR community. In this section, along with design goals, I will discuss representative works for creative media authoring in the context of direct manipulation user interfaces, tangible user interfaces, and sketch-based user interfaces.

2.3.1 Design Goal: High Ceiling, Low Floor

To date, most of the software systems for creative expression are either easy to learn (e.g., mobile paint applications), or extremely powerful (e.g., Photoshop, Illustrator, Power Point). Rarely are they both. Examining numerous audio-visual systems, Golan Levin suggested that an expressive audio-visual system should be "instantly learnable, infinitely masterable" [90]. Designing a media-authoring tool that is simple and instantly learnable and yet expressive and powerful is difficult to reconcile. Nevertheless, real-world exemplars of systems like *sand animation* and *pen illustration* come close to meeting this goal. For example, anyone can discover the basic principles for operation of a guitar easily; yet it is so expressive that a master artist can spend a lifetime practicing this medium. *Easy to learn, yet extremely powerful* as a goal for an interactive digital art medium is inherently contradictory. It is, notwithstanding, one of the essential goals of this thesis. Within the computational tools, Papert argued that programming languages for all should have "low floor and high ceiling" [112]. In addition to that, languages and medium often need "wide walls" so that they can support many different types of visual artifacts by different types of people. Satisfying the triplet of low floor, high ceiling, and wide walls is extremely challenging [124]. The design of SCRACH, a programming language for children, pursued low floor and wide walls [124], enabling children to create a wide range of interactive contents. Popular professional graphical tools (e.g., Photoshop, Illustrator, and Flash) pursued wide walls and high ceiling. In contrast, the projects within this thesis pursue low floor and high ceiling, to connect both professionals and amateurs to a very specialized and specific creative process.

2.3.2 Direct Manipulation and Tangible User Interfaces

Computational craft is enjoying a renaissance, with the rapid penetration of digital interfaces into every sphere of human life. The ubiquity of tablet computers, smart phones, and increased hardware capabilities has spawned numerous software tools with direct manipulation user interaction techniques. The intention of direct manipulation is to allow a user to manipulate objects presented to them using actions that correspond, at least loosely, to the physical world. Shneiderman [144] constructed an "integrated portrait" to define the characteristics of direct manipulation interfaces, involving continuous representation of objects of interest and rapid, reversible, and incremental actions and feedback. Direct manipulation interfaces are beneficial to users in a number of ways [144]. The real-world metaphors for objects and actions can make it easier for a user to learn and use an interface, and rapid, incremental feedback allows a user to make fewer errors and complete tasks in less time, as the user can evaluate their actions and outcome.

In the context of creative media authoring, a number research prototype systems explored the notion of direct manipulation interfaces to bridge the gap between the physical and digital world for painting and animation. In recent years, researchers have produced 2D animation by demonstration systems that could be used for digital art. K-Sketch [34] is a general purpose and informal sketch-based 2D animation tool that allows novices to create animation quickly and easily. But, in K-Sketch, all the interactions must be done through a single point. Researchers have also explored the use of multiple touch points to record real-time deformation of characters [57]. As for painting interfaces, Project Gustav [107] attempts to create a realistic painting experience while bridging the gap between the physical and digital world. Fluid Paint [158] and IntuPaint [159] provide a similar experience, using the entire region of contact between brush and surface to model brush strokes.

During the 90s, Hiroshi Ishii introduced the concept of *tangible bits* [61], computer interfaces providing physical representation and manipulation to digital bits. One of the key affordances of tangible user interfaces is the ability to employ physical skills for digital information manipulation. Researchers have also produced a number of notable tangible user interfaces for content creation. A notable painting system representation tangible user interface is I/O Brush [128], which allows artists to paint with patterns and movements "picked up" from everyday materials. I/O Brush looks like a regular physical paintbrush but has a small video camera with lights and touch sensors embedded inside. The artist can paint with the special ink in the canvas "picked up" by the I/O Brush. In Video Puppetry [11], artists record simultaneous manipulations of multiple physical puppets to create animation.

SandCanvas (Chapter 3) is a multi-touch digital art tool inspired by sand animation with direct manipulation capabilities. SandCanvas captures the entire region of contact between the surface and the artist's hand to model interactions with sand. SandCanvas bears a resemblance to the direct manipulation and tangible user interface art creation systems presented here. It is a medium for performance art where the final performance is a kind of 2D animation. Instead of animating a fixed or predefined set of characters or objects, however, the artist creates characters in sand using rich gestures that cannot be represented adequately with a set of discreet points. SandCanvas also has unique creative tools like recorded gestures that are not found in existing direct manipulation user interfaces for animation. Similar to other tangible user interfaces, SandCanvas leverages the delicate mechanical structure of human hands and skills for content creation.

2.3.3 Sketch-based User Interfaces

Sketching allows people to visually represent ideas quickly, without committing to decisions prematurely. Apart from its role for visual illustrations and drawings, sketching is widely adopted as a powerful tool for communication, visual thinking, and rapid design, due to its minimalistic yet greatly expressive nature [74]. Ever since Sutherland introduced the Sketchpad concept in the 1960s [56], sketch-based interaction has been extensively studied. Given the central role of sketching in design process and visual thinking, previous research on sketch-based interaction has infused a wide range of graphical applications for supporting pre-productive, exploratory activities in variety of domains and applications.

The strength of sketching input lies in the speed and fluidity with which people can express and modify shapes and relationships of the drawn objects without attending to details. However, the imprecise and ambiguous nature of sketching makes it difficult for computers to recognize it. The primary research themes in sketch focus on *traditional sketching* (sketching for design and early exploration), *hardware* (display and sensing technologies), *sketch recognition* (how to recognize and what to recognize), and *human-computer interaction* (interface widgets, design heuristics and interaction idioms) [74].

The projects presented in this thesis focus on *traditional sketching*, that is, sketching for computational design, rapid exploration, and brainstorming. The design of these projects capitalizes on the freeform nature of sketch-based interfaces for texture illustration (Vignette, Chapter 4), animation authoring (Draco, Chapter 5), and storytelling with data (SketchStory, Chapter 6). I will discuss representative works of sketch-based interfaces in the domain of static artifacts, animation, and visualization.

Sketch-based Interfaces for Static Artifacts

Much of the works of computer support for sketching have roots several decades ago. Sketchpad [56] was the first to demonstrate sketch-based user interface for human-computer interaction and computational design. Sketchpad was an interactive design system allowing engineers to create models and add constraints within those models by drawing with a light pen on a graphical display. RAND's GRAIL [42] system (GRAphical Input Language) interpreted sketches as a visual programming language for creating flowcharts. Later, sketch-based applications and interaction techniques were explored by researchers in a number of domains, including communicating early design ideas [45], 3D graphics modeling [129, 59], animation authoring [34], interactive interface prototyping [68, 88], and in special purpose applications such as MathPad^2 [72] and VectorPad [17].

In the domain of 3D modeling, Teddy [59] is a sketching interface that enables users to construct 3D models by freeform sketching. The system constructs plausible stuffed 3D models from freeform sketches and enables a number of manipulation functionalities with sketching. ILoveSketch [129] is a 3D sketching system that enables professional industrial designers to create 3D curve models.

Researchers have also explored sketching interfaces for prototyping. SILK [88] (Sketching Interfaces Like Krazy) allows designers to sketch user interfaces and storyboards and then interact with them. The system recognizes sketches of a limited set of common user interface elements (e.g., buttons, scroll bars) and then transforms the user interface into a high fidelity version. Similarly, DENIM [68] enables users to prototype websites and individual page layouts.One important property of these tools is that they support capturing and retrieving design histories of drawn objects, which is often desirable to designers to reflect their process.

Sketching interfaces have also been explored in a wide range of specific purpose applications, such as problem solving. MathPad^2 [72] and VectorPad [17] let students draw pictures of natural phenomena and relate them to equations. The equations in the systems govern the quantitative aspects of the drawings.

Project Vignette (Chapter 4) presented in this thesis is a special purpose application of sketch-based interface that enables users to create rich pen-and-ink illustrations quickly and easily. The key idea of this interface is the tight integration of texture synthesis algorithms with freeform sketching interaction techniques. In Vignette, the user sketches a small fraction of a texture and gestures the progression of the texture. The system then completes the texture by synthesizing from example strokes. This user interaction approach reduces tedium, yet preserves the expressiveness of the illustrations.

Sketch-based interfaces for Animation

Researchers have explored methods for easy animation authoring for novice animators using motion sketching [34, 57, 136, 154]. In motion sketching systems like K-Sketch [34], the animator can select an object and sketch the path for the object to follow [103]. In similar spirit, Popović et al. [115] explored the use of motion sketching with underlying physics simulation. In their approach, an animator sketches how objects should move and the system computes a physically plausible motion that best fits the sketch by estimating the best physical parameters with optimization algorithms. Other tools exploit motion sketching for specific purpose animations, such as character movement [154]. DirectPaint [136] examines pen-based techniques to edit visual attributes of moving objects along their trajectory, consolidating spatial and temporal controls. Common among these systems is that they allow animation for only a single object at a time, therefore requiring numerous iterations to animate a whole collection. Furthermore, these systems lack high-level controls to tune the collective and individual behavioral properties of numerous elements.

Vignette (Chapter 4) allows users to efficiently brush textures and collections of objects, but those objects did not move. Draco (Chapter 5) provides a similar interaction metaphor, but expands it to support texture motion by allowing users to efficiently specify animations for collections of objects and subsequently adjust the properties of the global animation as well as finely tune the granular motions of the individual objects.

Sketch-based Interfaces for Communication and Visualization

In the context of InfoVis, sketch-based interaction was first used to support data queries with sketches. For example, QuerySketch [95] and QueryLines [77] enable people to specify queries of time-series data by drawing a freeform line graph as a target pattern. Recognizing the benefits – promoting thinking, insight, and inspiration – of the act of sketching [94], the InfoVis community has recently started to employ sketch-based interaction for data exploration. For example, NapkinVis uses pen gestures to support fast and effortless visualization construction [162], and SketchVis leverages hand-drawn sketch input to quickly explore data in simple charts without using menus or widgets [63]. To further advance these approaches, Walny et al. investigated the use of pen and touch for data exploration on interactive whiteboards [70]. Their study on the distinctive role of pen and touch interaction shows that people can transfer knowledge from interaction with the physical world, leading to more natural and learnable interaction techniques.

SketchStory (Chapter 6) leverages the expressiveness and freeform nature of sketch for the creation of interactive and organic charts for narrative storytelling with data. Previous research shows that, when both pen and touch interactions are supported, people clearly distinguished between appropriate pen and touch interactions. SketchStory distinguishes the role between pen and touch interactions to avoid having two explicit modes for chart creation and management; using the pen for drawing charts or annotations and touch for manipulating them. For example, the presenter can move visual elements (e.g., charts, selected group of strokes) with touch without explicitly changing the mode.

Vignette, Draco, and SketchStory: Sketching + Synthesis

The three sketch-based projects (Vignette, Draco, and SketchStory) presented in this thesis seek inspiration from the existing works in this domain, yet distinguish themselves with unique affordances and by incorporating powerful end-user programming capabilities by synthesizing from example sketches. These projects take sample input from the users and the system performs the repetitive work by:

- Auto-completing the textures from example strokes by spatial texture synthesis (Vignette, Chapter 4).
- Creating animations of groups of objects by spatio-temporal synthesis (Draco,

Chapter 5).

- Creating non-photorealistic sketchy rendering charts by synthesizing the icons with the underlying associated data (SketchStory, Chapter 6).

2.4 Research in Stylized Graphical Rendering

Stylized graphical rendering, or non-photorealistic rendering (NPR), is an area of computer graphics that focuses on enabling a wide variety of expressive styles for digital art. In contrast to traditional computer graphics, which has focused on photorealism, NPR is inspired by artistic styles such as painting, drawing, technical illustration, and animated cartoons. The related work in non-photorealistic rendering (NPR) inspires the works in this thesis in important ways. Prior works in this field aids the algorithm developments and simulations to reproduce physical drawing styles into digital media – such as sand rendering and generating pen-and-ink style textures.

In the early days of computer graphics, researchers thrived towards realism and photorealistic rendering for visual imaging. Computer graphics research and pipelines were geared towards physical accuracy and realism [106, 108, 113, 121, 122, 125]. Tools and techniques were tailored for professional artists in the context of production applications. In the 1990s, computer graphic researchers started to explore rendering techniques for visual imaging that are not photorealistic, rather inspired by non-photorealism and traditional artistic styles. As defined by Holger, non-photorealistic rendering (NPR) is *computer-enabled synthesis and tools for art creation and reproduction* [168]. Ever since, NPR techniques have been widely applied and explored in a range of domains. In this section, I will briefly discuss representative works of NPR in art, entertainment, visualization, and presentation.

2.4.1 NPR for *Art Creation Tools* and *Style Reproduction*

NPR has gained significant attention in artistic tools and production environment in a number of ways. There are *artistic creation tools* that mimic real world physical tools (such as paint brushes [107]), but are often augmented with unique digital affordances, such as undo, copy/paste, parameter controls, etc. In most of these cases, these tools rely on highly realistic, physical or empirical simulations. A physical brush consists of thousands of individually deforming bristles interacting with viscous fluid paint and a rough-surfaced canvas to create rich, complex strokes. An example of an NPR *artistic creation tool* is Project Gustav [107], a digital painting-system prototype that enables artists to create realistic brush strokes by leveraging the power of GPUs and natural media-modeling and brush-simulation algorithms.

The other use of NPR is automatic or user-assisted *style reproduction* [168]. Style reproduction NPR tools facilitate cheaper, efficient, and convenient ways of content production that preserves a certain artistic style (e.g., pen-and-ink illustration, cartoon style or watercolor). For example, automatically creating a live-action video into a cartoon style rendering [86, 110, 170] can significantly save an artist's time and production cost. Another example of *style reproduction* is pen-and-ink illustration rendering. A number of systems render illustrations, 2D images, or 3D models in a pen-and-ink style. Some are geometry-based [133, 166, 167], taking 3D scene descriptions as input, while others take 2D images as input [37, 134]. The tones and textures of the resulting ink illustration from these systems are therefore guided by the underlying 3D geometry or 2D image.

Given the emergence of NPR in *art creation* and *style reproduction* tools, both the movie (especially animated cartoons) and computer games industry embraced non-photorealistic rendering techniques as a valuable technical resource and differentiator. Typically, animated movies require 12-24 frames per second. As such, NPR style reproduction tools became an invaluable resource for such production to save time, reduce production cost, increase throughput, and facilitate artistic experiments. Cartoon is one of the overwhelmingly prevalent and popular NPR styles, both in films and games. Researchers have produced a wide range of cartoon rendering approaches [86, 110] with stylistic extensions and variations [169]. In the context of gaming, early computer games were more abstract due to limited hardware capabilities. As the graphics hardware became more powerful, the games pushed towards more realistic rendering. At that point, several game studios started turning to more stylistic NPR rendering to differentiate them from the competitors. Examples include *Prince of Persia* (Ubisoft, 2008), *Borderlands* (Gearbox, 2009), and *Limbo* (Playdead, 2011), among others.

The projects in this thesis fall somewhere in between NPR *artistic creation tools* and *style reproduction* tools. SandCanvas (Chapter 3), Vignette (Chapter 4), Draco (Chapter 5), and SketchStory (Chapter 6) provide style specific tools, similar to NPR *artistic creation tools*, enabling users to start completely from scratch and preserve expressiveness. But, at the same time, these tools reduce tedium by synthesizing from user-defined examples, bearing similar motivation to NPR *style reproduction* tools.

2.4.2 NPR for Visualization and Presentation

Apart from the artistic tools and entertainment production, non-photorealistic rendering (NPR) styles have been explored extensively to illustrate 2D shapes and 3D objects across a wide range of domains for practical and aesthetic advantages [69, 71]. Traditional architectural designs, for example, start from initial sketching and then gradually evolve to final polished 3D renderings. During the early stages of architectural design, the goal is to make informed decisions and brainstorm without overwhelming the clients with superfluous details that distract from high-level design for decision-making. In the seminal paper by Schumann et al. [69], they performed an empirical study involving 54 architects to compare the standard CAD software output with sketch-renderer for several architectural scenes. Schumann et al. found that the stylized rendering was preferred in the early phases of design as it was visually more engaging. They also elicit participation, greater clarity, and active discussion. More recently, Core et al. [27] conducted a study showing that people can interpret shapes accurately when looking at drawings, for drawings made by both artists and computer graphics algorithms.

In addition, NPR techniques are found to affect viewers emotionally [105]. More recently, Wood et al. [71] demonstrated the use of NPR specifically in statistical data visualizations and hypothesized about its potential role in constructing visualization narratives. Compared to traditional data visualizations, their study indicated increased engagement and active participation with NPR data charts. The ability of NPR to evoke emotional response and to provoke active participation and distinct visual appeal is encouraging to our work as we pursue similar goals with our narrative storytelling tool. In this thesis, SketchStory (Chapter 6) employs integration of sketching and data chart rendering, where the sketched input is used for rendering the data charts.

2.4.3 NPR Tools and User Engagement

Computer graphics researchers, in particular the NPR community, had a considerable impact in art creation tools, production, design, and presentation. However, these application areas are almost exclusively limited to professional use. In addition to that, much of the academic NPR research in *style reproduction* tools is geared towards automated solutions, reducing the role of artists for digital content creation. For instance, tools developed for pen-and-ink style renderings [10, 37, 133, 134, 166, 167] require some kind of 3D models or 2D images to serve as the template for guiding the generation of textures. As a result, these tools can create high quality pen-and-ink style drawings, but there is little room for variation and artistic styles. In contrast, traditional pen-and-ink illustration accommodates a wide range of artistic styles, which make it a very popular medium. However, recently there is a growing interest for NPR application for casual users with increased user involvement in content creation. Involving the users in content creation is important for various reasons [168]. It provides more artistic freedom and enables users to incorporate their personal style into the final artifacts, which could never be possible with automated solutions.

Additionally, a user can provide perceptual or semantic input that significantly improves the results of NPR algorithms and tools.

The projects presented in this thesis, SandCanvas (Chapter 3), Vignette (Chapter 4), Draco (Chapter 5), and SketchStory (Chapter 6), bear similar motivation for casual creativity through increased user engagement and assistance. Much of the academic NPR work has emphasized the final visual artifacts with semi-automatic approach. But, from a design perspective, it is imperative to look at the process and workflow of creating those artifacts. The digital tools presented in this thesis combine the best of the digital and physical worlds by infusing and augmenting traditional process of creating artifacts a with digital power.

2.5 Designing Digital Art Media

> *At the heart of any craft practice, lies the idea of applied, skilled understanding and mastery of material* – Scott McCloud [101].

In his seminal book *Understanding Comics* [101], McCloud illustrates an in-depth understanding of the comic media and how to effectively use the constraints and unique affordances of the media to tell a story. Similarly, I believe the key to digital art design is to have an applied and skilled understanding of the medium and see the different aspects of the medium with sufficient clarity. The moment a designer clearly understand an art medium – the needs, affordances, desires, or artists involved – then he or she can tackle it. New media can come from a leap based on observation and experience of existing media.

The design process of the projects in this thesis starts with a traditional medium,intending to understand the unique affordances of the medium and materials, the artifacts (visual effects, techniques, and styles), and the creation process. Prior descriptive art literature of the related medium aided the understanding during the design process. The insights aided the development of features and workflow. For instance, while designing Vignette, we identified three major types of textures in existing pen-and-ink illustration drawings. As such, we provided three synch texture synthesis rules – brush, hatch, and fill. Importantly, the design process does not simply reproduce an existing art medium into digital space, rather the new affordances and digital capabilities make the resulting medium a new and unique one. For instance, the design of SketchStory was primarily inspired by whiteboard animation. But, the unique affordances of SketchStory, such as the data driven chart synthesis techniques and interactivity, distinguish it from traditional whiteboard animation techniques.

However, in general, design process cannot be reduced to a formula. While researchers have extensively explored art analysis, interaction design (sketch-based interfaces and direct manipulation interfaces), and artistic (non-photorealistic) rendering, the design of the computational tools in this thesis takes a holistic approach to understand the process, artifacts, and medium. Chapters 3-6 explain this design process, implementation, and user evaluation of these individual projects in depth, illustrating how digital art and communication media design can be inspired by rich and diverse sources of information.

Sir Ken Robinson pointed out the role of technology for creativity:*The tools themselves are always neutral. They rely on the intentions of people. It's all about the possibilities people see in them and the opportunities the tools provide for imaginative work* [173]. Technology can foster digital art in many ways. With the ubiquity of tablets, computers, and increased computational power, advances in technology are allowing more people to explore art and content creation than ever before, instrumenting the democratization of creativity [173]. Additionally, it also creates enormous opportunity for collaboration in media authoring and creative endeavors. Social media and networked tools make available mental collaboration on a scale that is not possible with traditional art media. Another role for digital art tools might be to augment an artist's ability to consider more possibilities, where artists can seek out many possibilities with remarkable speed. Great artworks are driven by more explorations by artists. The collaborative and exploratory aspects of digital art remain as future work.

3. SANDCANVAS: A MULTI-TOUCH ART MEDIUM INSPIRED BY SAND ANIMATION

Figure 3-1: A user interacting with SandCanvas (left), and images created with SandCanvas (right).

Chapter 1 introduced the overarching goal for the design of new media arts tool, and Chapter 2 presented the related work in understanding art, interaction design, graphical rendering and re-emphasized the need of holistic approach for design. In this chapter, we will switch to an individual and exemplary case study for developing a new digital art media. As a starting point, we will consider sand animation.

3.1 Background and Motivation

Sand animation, also known as sand art, is a form of visual storytelling in which an artist dexterously manipulates fine granules of sand to produce images and animations. The process begins by applying sand to a lighted surface, after which images are rendered on the surface by drawing lines and figures with bare hands. It is an increasingly popular medium for performances and stop-motion animation [82].

Two characteristics combine to make sand animation a unique art form. First, because it is a performance medium, its attraction and aesthetics are closely tied to the creation process as well as the finished artwork [87, 91]. The creation process in performance media is improvisational, fast, continuous, and often accompanied by other forms of performance art, like music, choreography, drama, and dance. The sand animator's task is to unfold a narrative through a progression of visual images produced with a seamless stream of physical gestures.

Second, sand animations are formed through a powerful and expressive vocabulary of physical interactions between artist's hands and small granules of sand. In contrast with sketches or paintings, which are produced with discrete pen or brush strokes, sand animation leverages the delicate structure of the artist's whole hand (often both hands). These hand gestures are easy to learn, quick to perform, and economical to correct, which makes this medium suitable for exploration and brainstorming in addition to storytelling through live performance.

Sand animation has increasingly attracted audiences and artists because of its innovative and expressive graphic style [82, 126]. However, sand-animation performance spaces are difficult to set up and maintain [135], which prevents many novices from getting started. This led us to create SandCanvas, a new digital artistic medium inspired by sand animation. SandCanvas adds undo and recording features that make sand animation easier to produce, it allows easy experimentation with colors and textured backgrounds, and it adds new capabilities that go beyond traditional sand animation, such as recorded gestures and video mixing.

The increased availability of multi-touch display surfaces has removed some of the technical obstacles to creating a digital sand animation medium, but we still faced significant challenges. Multi-touch UI toolkits do not currently capture all the richness in human hand gestures [102]. In particular, they do not attempt to map touch regions in the current time step to touch regions in the previous time step. This required us to devise a new, fast approach to performing this mapping. Also, simulating the physical behavior of sand in real time is still a major challenge. We achieved real-time performance by optimizing an existing technique [150] and using graphics hardware acceleration.

This project makes the following contributions:

- We introduce a new digital artistic medium that leverages the expressiveness of hand gestures on a multi-touch platform to provide a visual experience that goes beyond physical sand animation.

- We analyze the sand animation process, highlighting common pouring and manipulation techniques and developing a taxonomy of hand gestures.
- We present SandCanvas's intuitive UI design and its enhanced digital capabilities.
- We describe our implementation, which employs new techniques for performing real time sand simulation in response to gestural input.
- We evaluate SandCanvas with 1 professional artist, 4 amateur artists and 2 novice users to gain insight into the importance and unique affordances of this medium.

3.2 Technology for Performance Art

New media interactive installations and recent advances in interactive surfaces inspired the development of SandCanvas. Direct manipulation and tangible user interfaces for digital art are discussed in Chapter 2. In this section, describe related systems and algorithms for sand motion simulation. We then discuss systems closely related to SandCanvas.

3.2.1 Technology for Performance Art

Performance art has a rich history that spans hundreds of years [90]. Myron Krueger's Videoplace, developed between 1969 and 1975, was an early interactive artwork that incorporated computer vision [85]. Since Videoplace, numerous audio-visual performance systems have been driven by human gestures [91, 147]. Examining numerous audio-visual performance systems, Levin derived a set of design goals for new performance art systems [91]. According to him, successful systems should be predictable, instantly knowable and indefinitely masterable. We pursued similar goals when designing SandCanvas.

3.3 Physical Sand Simulation

Since the sand particles used in sand animation are very fine, the number of sand particles is potentially huge. Hence, physically accurate interaction with sand is particularly challenging. Li and Morshell devised one simulation approach, but it assumes that sand is moved by convex objects only [92], which prevent touch regions of arbitrary shape from interacting with sand. Bell and colleagues devised a sand simulation method that handles arbitrary shapes [15], but it models each grain as a discrete element and will not produce real-time simulations on the scale needed for sand animation.

Summer and colleagues developed a faster technique that still falls short of real-time performance [150]. Onoue and colleagues sped it up by assuming that only rigid objects would interact with sand [111]. Our method is also based on Summer's, but we do not assume objects are rigid, because an artist's hand can change as it moves across the canvas.

3.3.1 Sand Art for Storytelling

We have found few sand art systems worth noting. Hancock and collegues' sandtray therapy system allows storytelling on a sand background, but users manipulate figurines instead of sand [49]. Ura and colleagues developed a tool for painting with simulated sand, but it reduces input to discrete points [157]. iSand is an iPhone application for sand art that shares this limitation, and its sand granules are much larger than those used in traditional sand animation.

In contrast, SandCanvas captures rich human hand gestures in multiple areas instead of multiple points. It also preserves the expressive and playful nature of sand animation and adds new capabilities that go beyond traditional sand animation, such as recorded gestures and video mixing.

3.4 Sand Animation Analysis

To better understand the special requirements of our new medium, we analyzed the medium, process and artifacts of traditional sand animation. We enumerated the gestures commonly used by sand animators based on our analysis. After a meticulous observation of 30 sand animation videos, we identified a set of sand animation techniques commonly employed by artists.

3.4.1 Sand Animation: Medium

The fluid properties of sand, unique graphic style, and real-time content creation make sand animation a unique form of performance art. Since, it is a performance art, the aesthetic of the medium largely relies on how contents are being created in real time. One of the pleasant properties of sand animation is how artists fluidly evolve one scene into another one, surprising and delighting the audience (**Figure 3-6**).

3.4.2 Sand Animation: Process and Artifacts

Sand animators can create a wide range of visual effects using hands only, without additional prosthetics. We analyzed the range of visual effects created by artists, and captured the process of creating those effects. We found two major modes of techniques in sand animation, pouring and manipulation.

Pouring Techniques

Pouring is an additive technique that varies depending on how much of the canvas is affected. Canvas pouring is used to set the texture and initial context for painting (**Figure 3-2** left), or, to change context while storytelling. Skinny pouring is used to draw tiny details, lines, and shapes (**Figure 3-2** right).

Manipulation Techniques

Sand manipulation techniques move sand rather than adding it. We classified these techniques by how the artist's hand interacts with sand. Fingertip drawing traces out lines with the tip of one or more fingers (see **Figure 3-3** left). While, finger carving (see **Figure 3-3** right) uses the whole finger, typically the index finger, small finger, or the outside of the thumb, for drawing and fine-tuning shapes.

Artists do not use their fingers exclusively. Palms are often used to create semi-elliptical, or spiral like patterns, such as clouds. We call this technique palm rubbing (see **Figure 3-4** left). Whole hands are often used to make big sweeps to clear the canvas and set up a new context for the animation, which we termed hand sweeping (see **Figure 3-4** right).

One final technique that bears mentioning is actually a special version of other techniques. Sand animators will sometimes use both hands simultaneously to quickly draw or pour symmetrical patterns in sand (see **Figure 3-5**). This technique, which is quite rare in other artistic media, is very common in sand animation.

These techniques can be combined to fluidly transform one image into another (see **Figure 3-6**), creating surprise and conjuring emotion. Here lies the beauty of sand animation.

Pouring	**Manipulation**
Skinny	Palm rub
Canvas	Finger tip
	Finger Curve
Symmetrical	Hand Sweep

Table 1: Common sand animation pouring and manipulation techniques. *Symmetrical* is a modifier that can apply to both pours and manipulations.

Figure 3-2: Canvas pouring (left) creates background textures, while skinny pouring (right) is for drawing lines.

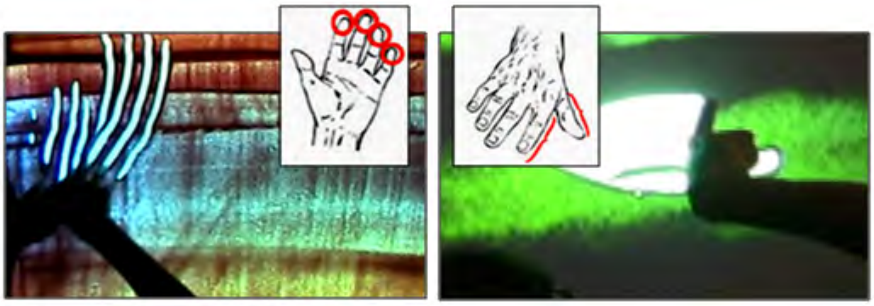


Figure 3-3 *Fingertip drawing* (left) and *finger carving* (right) to create and manipulate shapes.

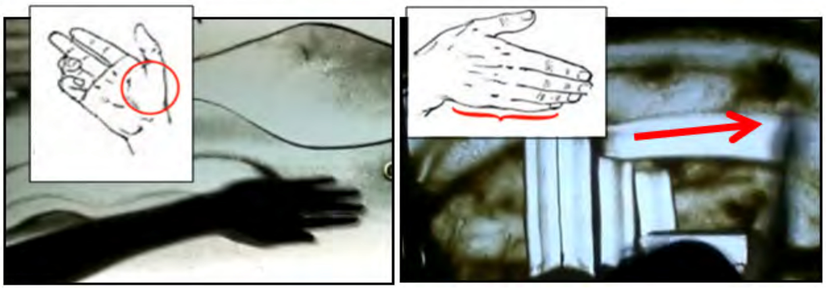


Figure 3-4: A palm rub (left) draws cloudy patterns, and a *hand sweep* (right) clears part of canvas.

Figure 3-5: *Symmetrical hand sweep* (left) & *skinny pour* (right).

Figure 3-6: Fluid transformation of images.

3.4.3 Taxonomy of Sand Animation Gestures

After listing common sand animation techniques, we saw many similarities and differences between them. To better compare and contrast these techniques, we created the low-level taxonomy of gestures found in Table 2. While there are other gesture taxonomies in the literature [39, 172], we needed one that was created specifically for multi-touch art work like sand animation.

We manually classified gestures along five dimensions: mode, form, precision, hands and actuation. Mode separates pouring gestures from manipulation gestures. Form indicates any motion in the gesture. In static gestures, the hand is held in one position and one configuration, while dynamic gestures change the position or configuration of the hand. (This is similar to the pose and path concepts in Wobbrock et al.'s gesture taxonomy [172]) The precision of the gesture can be coarse or fine, and the hands dimension indicates the number of hands involved in a gesture: one (uni-manual) or two (bi-manual).

Finally, actuation indicates the portion of the artist's hand that interacts with sand: a single finger, multiple fingers, the palm (without fingers), or the whole hand (both palm and fingers). When using one or more fingers, we distinguish between the finger tips and the side of the finger. We also note when artists use tangible objects to interact with sand.

This analysis of sand animation gestures helped us to understand the range of interactions that sand animators need in SandCanvas. The following section explains how we designed and developed a UI to support these gestures

Categories	**Categories**	**Description**
Mode	*Pouring*	Pouring on surface
	Manipulation	Manipulating on surface
Form	*Static*	Hand held in one position
	Dynamic	Hand moved
Precision	*Coarse*	Gesture has low precision
	Fine	Gesture has high precision
Hands	*Uni-manual*	Use one hand
	Bi-manual	Use both hands
Actuation	*Single fingertip*	Use single fingertip
	Single finger side	Use side of a single finger
	Multi-fingertip	Use multiple fingertips
	Multi-finger side	Use side of multiple fingers
	Palm	Use palm without fingers
	Hand	Use both palm and fingers
	Tangibles	Use other objects

Table 2: Taxonomy of sand animation gestures. The underlined letter(s) in each category indicate the abbreviation used for that category in later tables.

3.5 SandCanvas: Design and Implementation

SandCanvas is designed to run on an interactive surface based on the principle of diffused illumination [1]. Our table uses a 29cm by 21cm white acrylic surface as a diffuser and projection screen for a rear-placed LCD-projector. An array of 140 infrared LEDs also shines on the surface from below, and objects touching the surface reflect this infrared light back on a 320 by 240 pixel infrared video camera. Our software was written with OpenFrameworks, a C++ toolkit for graphic applications with image processing tools. This software runs on a 3.0 GHz Intel Core2 Duo CPU E8400 running Windows Vista with 4GB RAM and a graphics card with an nVidia GeForce 9500 GT2 GPU.

3.5.1 Sand Simulation

Real-time simulation of sand movement in response to rich hand gestures is a challenging problem. Here we describe all the steps in our simulation process.

Tracking and Modeling Contact Shapes

As our multi-touch platform was vision-based, we used standard image processing techniques for contact shape detection. We used dynamic background subtraction to remove the background from the current frame, thresholding to adjust the level of acceptable tracked pixels, Gaussian blur for smoothing and filtering out random noise, and highpass filter to amplify edges. The resulting contact shapes are represented as 2D polygons. We then use the Community Core Vision tools to correspond touch regions with one another across successive frames.

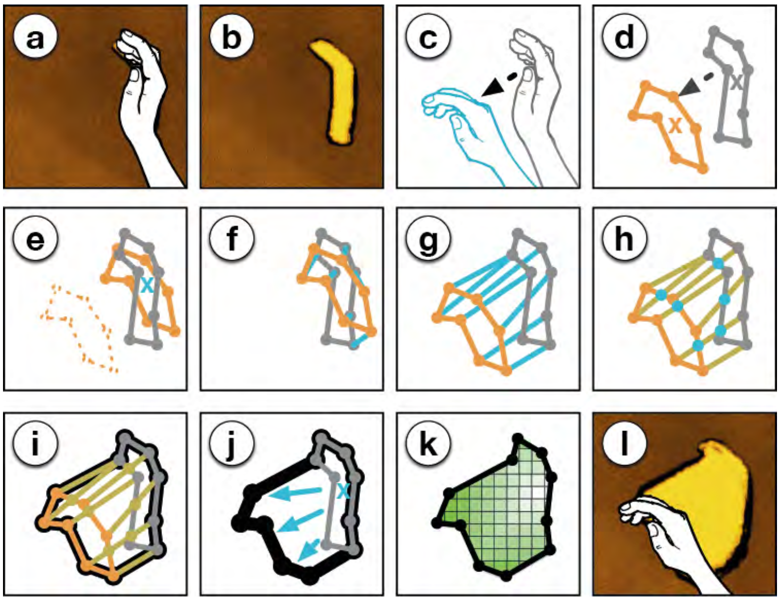


Figure 3-7: A step-by-step illustration of computing the collision region and pushing the sand. (a) The hand first touches the surface. (b) The sand is pushed to the boundary of the contact shape. (c) The positions of the hand in two successive frames. (d) Their contact polygons. (e) The two polygons are aligned at their centroids. (f) Point-to-point correspondences are computed between the two polygons. (g) A graph is constructed consisting of the polygons and the correspondence edges. (h) The intersections of the edges are computed. (i) The boundary of the whole graph is computed. (j) Sand is to be pushed from the grey frontier to the black outline. (k) A distant map is constructed and sand is pushed from the lighter pixels to the darker pixels until it reaches the boundary of the collision region. (i) Final rendering of the new sand height map (after some sand erosion).

Sand Modeling

Although particle systems and voxels are commonly used for modeling the motion of granular materials, they are computationally expensive and cannot handle a very large number of sand granules. Instead, SandCanvas uses a discrete height field that is often used to model ground surfaces [111, 150]. The height field has a resolution equal to the screen resolution, thus, each pixel has a height value (16-bit float), which we call a column.

Given a set of hand contact shapes and a grid of sand columns, we compute the sand deformation in three steps:

- First, when a contact shape moves on the surface, we find the columns of sand that collide with the path of this contact shape.
- Second, sand within those columns is pushed outward towards the surrounding columns.
- Finally, by detecting steep slopes, sand is moved from higher columns to lower columns, producing realistic sand settling motion or sand erosion.

Computing the Collision Region

When a contact shape first touches the surface, the collision region is the polygon that represents the contact shape itself. However, when a contact shape moves on the surface (**Figure 3-7**(c)), the first part of our algorithm computes the region swept by the contact polygons across successive frames. Sand within this collision region need to be pushed on the surface. **Figure 3-7**(d) shows the contact polygons for the current frame (f_t+1) in orange and the previous frame (f_t) in grey. We must now compute the collision region from these two polygons.

Our first step is to compute the point-to-point correspondences between the two polygons. An iterative-closest-point method [16] could be used to compute these correspondences, by rotating the polygons until they are closely aligned. While this gives accurate results, the iterations take a long time. Instead, we simply align the centers of the two polygons and examine each point in the f_{t+1} polygon to find its closest neighbor in the f_t polygon (see **Figure 3-7** (e–f)).

In our second step, we construct a graph, which consists of the polygon in f_{t+1}, the polygon in f_t, and the correspondence edges (see **Figure 3-7**(g)). We then compute all the line segment intersections and create a DCEL (doubly-connected edge-list) for this graph (see **Figure 3-7**(h)). Finally, we compute the outline of the graph by finding the lexicographically minimal point and walking along the outside edges until we reach our starting point (see **Figure 3-7**(i)).

Volume-Conserving Sand Displacement

Now that we have found the columns that collide with the user's hand, we need to push the sand (as shown in **Figure 3-7**(j)) from the previous frontier (grey edges) towards the new frontier (black edges). To do this, we calculate the Euclidean distance transform within the collision region from the center of the previous polygon f_t (this center is the "x" in **Figure 3-7**(j)). In the resulting distance map (illustrated in **Figure 3-7**(k)), for pixels surrounding the collision region, their distance values are set to a very high distance value (higher than any value computed in the distance transform). To propagate the sand, the sand in each column within the collision region is evenly distributed to its neighboring columns that have higher distance values. We use the algorithm in [139] to perform the distance transform and the propagation efficiently on the GPU, which we implemented using OpenGL Shading Language fragment shaders.

Sand Erosion

Our method to simulate the settling motion of the sand around the collision region is based on the algorithm described in [150]. Our method is implemented for the GPU using OpenGL Shading Language fragment shaders. In the first pass, the slopes of each pixel with the eight neighboring columns are examined. If a slope is larger than a threshold value, excess sand is distributed from the higher column to the lower column using a gathering approach in the second pass. The two passes are performed every rendering frame, and the sand erosion takes many frames to complete, producing the realistic effect of sand rolling down the slope over time. To produce the asymmetric erosion caused by the temporary obstruction of the hand, unlike [150], we simply do not distribute sand into any collision region that might exist during a frame.

Performance

Our system runs at interactive rates (20–35fps). For a single finger, the average fps is 35, while for 7–8 fingers manipulating at the same time, the average fps is 20–25. With these frame rates, sand movement does not significantly lag behind hand movements, and users are able to feel immersed in the sand animation experience.

3.5.2 Exploring the Design Space for Sand Pouring

In addition to developing the algorithm to simulate sand manipulation, another essential component of Sand Animation is sand pouring. Most sand animators begin new scenes in their animations by quickly pouring sand on the canvas to set a background texture. We considered using computer vision techniques to distinguish pouring from manipulation gestures, but we quickly determined computer vision was not up to the task, given the variety of gestures and lighting conditions. Building a 3D deformable mesh model of the user's hand would achieve the highest fidelity, but it is difficult to build this model in a robust way [165].

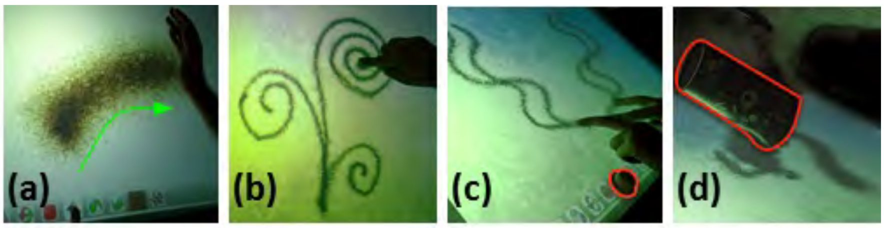


Figure 3-8: Pouring with touch and tangibles. (a) Canvas pour with whole hand. (b) Skinny pour with fingertips. (c) Symmetrical pour, tangible on button. (d) Pour with tangible.

Instead, we designed a bi-manual touch interface for pouring. Users touch a pouring button with their non- dominant hand while specifying a pouring region with their dominant hand. This style of interaction avoids mode error. It has been shown to be effective [93] and has been used successfully in several systems [34, 53]. Users distinguish canvas pouring from skinny pouring by the size and shape of the pouring region. Sweeping through the canvas with the whole hand results in canvas pouring (**Figure 3-8**(a)), while pointing or tracing a path with a fingertip results in skinny pouring (**Figure 3-8**(b) and **Figure 3-8**(c)). Users can place tangible objects on the pouring button if they wish to pour with both hands (**Figure 3-8**(c)).

We also allowed users to pour sand using tangible objects such as jars, thinking that this would provide a more natural feel (**Figure 3-8**(d)). In vision based multi-touch systems like ours, bright objects can be detected even when they do not touch the surface. We attached a piece of white paper to the face of a black jar so that it would pour sand when brought close to the surface in pouring mode.

3.5.3 SandCanvas User Interface

Before designing SandCanvas's user interface, we interviewed two professional sand animators to learn how they would like to enhance sand animation in the digital form. We first interviewed Sheh Meng, a professional practitioner with 10 years of experience in performing and teaching sand animation. According to him, most sand animators record video clips of their animations and edit them in a post-production step. Post production also allows animators to play with colors, saturation, and contrast. Hence, a desirable system should provide these capabilities. Sheh Meng also asked for features that allow new types of expression. He suggested a tool for recording gestures and saving them for future reuse.

Second, we interviewed Erika Chen, the winner of "Impresario the Open Platform" 2010. Erika is the world's first singer sand animator, having unique, extensive collaborations with drama, dance and live musicians. She was mostly interested in mixing sand animation with other media, such as clip art or ink drawings.

Our final user interface for SandCanvas is the toolbar shown in **Figure 3-9**. This toolbar appears at the bottom of the canvas, and it can be reduced to include only the sand pouring and expand UI buttons if the artist desires more canvas area. Based on our interviews, we put the following tools into this toolbar.

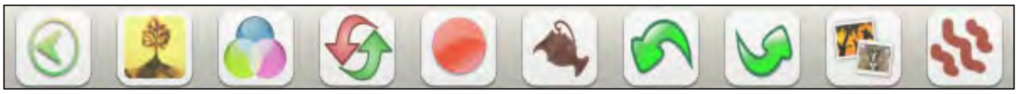


Figure 3-9: The user interface panel. From left - hide UI panel, change texture, change color, reset, record session, pour, undo, redo, capture frame and record gesture.

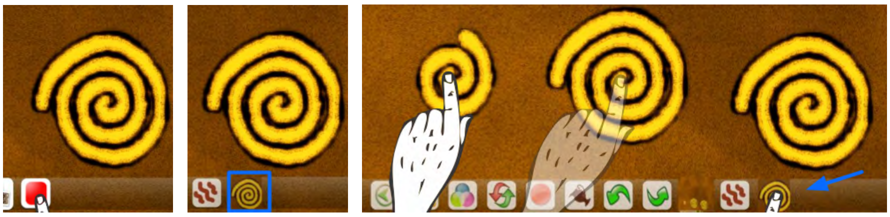


Figure 3-10: Steps of Gesture recording and playback after pressing the gesture record button (a) User draw a gesture and stop recording (b) An icon having the gesture appears in the UI panel (c) By pressing the recorded gesture button with non-dominant hand and touching by another hand initiates the gesture in different parts of the canvas.

Figure 3-11: Three key frames for stop-motion animation.

Record Session. Users can record their animation as video for later editing.

Undo and Redo. Users can undo and redo up to five operations. This number can be increased at the cost of additional memory.

Change Texture. Users can start with an empty canvas or they can choose from a set of predefined sand textures. Some textures are computer generated, while others are images of real sand.

Reset. Change the surface to the initial state of the texture.

Record Gesture. Users can press a button to begin recording a sequence of pouring or manipulation gestures. Pressing the button a second time stops recording and places a new gesture button icon in the toolbar (see **Figure 3-10**). The recorded gesture can be played back by touching this gesture button with one hand and touching the canvas with the other hand. Each touch plays the gesture starting at that touch point. This enables users to play gestures in parallel in different parts of the canvas.

Capture Frame. Users can capture snapshots of the canvas to use as frames in a stop-motion animation (see **Figure 3-11**). After pressing the capture frame button, it changes to show a thumbnail of the image that was captured.

Change Color. In film and animation production, color is used to create specific moods [8]. For example, a love scene will need different colors than a suspense scene. In SandCanvas, users can create a sequence of color gradients before a performance and cycle through them by pressing the change color button.

Enabling Mixed Media. In film production and storytelling, mixed media refers to the mixing of images from separate sources [8, 126]. SandCanvas allows users to define a set of still images and video clips that will appear underneath the sand during a performance. These images and clips are placed in sequence with color gradients and are also accessed through the change color button.

As soon as our system was implemented, we wanted to evaluate it to establish its usability and to understand the importance and the unique affordances of this artistic medium. The details of our evaluation are described below.

3.6 User Evaluation

Before our final study, we performed a pilot study with three users who gave us qualitative feedback, which we report here with other data. Our final study with seven users was both qualitative and quantitative, using a formal protocol designed to answer the following questions:

Q1. How do users evaluate the realism, fidelity and intuitiveness of Sand

Canvas?

Q2. Do users find the novel features of SandCanvas useful, and can they apply them effectively?

Q3. Can we gain further insight into the expressiveness of this new medium by analyzing gestures that users employ?

Q4. Is there evidence that SandCanvas facilitates creativity?

3.6.1 Participants and Environment

Our formal study had seven participants, all males ranging from 24 to 29 years old (M=26, SD=1.63). Among them, one is a professional artist, four are amateur artists, and two are novice users. 4 out of 7 users reported that they create artistic works once a week. Our pilot study participants were three females aged 26 to 29. One is a professional artist with prior sand animation experience and the other two are amateur artists. All evaluation sessions took place in a university laboratory using the tabletop system described previously in our design and implementation section. Lighting in the room was dim to give the surface maximal tracking accuracy. Each user received $25 for their participation.

3.6.2 Method

The formal evaluation process was conducted in the following four steps.

1) Exploration: 10 minutes. In this step, users were given no explanation of the system, and were told to play with SandCanvas while thinking out loud. This step helped us gauge the initial learnability of the system and users' initial impressions.

2) Training: 10–15 minutes. In this step, users were given a brief description and demonstration of the features they didn't discover in step one. We asked users to recreate a sequence of five drawings, each designed to teach sand animation techniques (**Figure 3-12**). Users first recreated all five drawings in their own way. After this, a facilitator demonstrated an easy way to create each drawing, and asked the user to try again.

Figure 3-12: Pictures given to users in step 2 (training).

3) Guided task: 10–15 minutes. In this step, users were asked to create an animation sequence based on three key frames provided (see **Figure 3-11**). This step allowed us to compare user performance on a fixed task.

4) Free task: Up to 30 minutes. In the final step, users were asked to use their own imagination and create the best sand animation they could. This step helped us assess users' preferred techniques, and it allowed us to observe creative use of SandCanvas.

At the end of the study, users were given a questionnaire and interview. The entire study took about 90 minutes.

3.6.3 Results and Discussion

Users' overall reaction was very positive. They found SandCanvas's UI intuitive and they were able to create meaningful artworks in the time they were given. The medium was a pleasure to use; as one user reported, "The ability to play with sand itself is the most interesting part."

Q1. How do users evaluate the realism, fidelity and intuitiveness of SandCanvas?

Most of our users felt that the behavior of virtual sand in SandCanvas closely mimics the feeling of physical sand. They often perform gestures on SandCanvas as if they were playing with real sand: 3 users piled sand in the middle of the canvas and observed its spreading behavior. In the post-study questionnaire, users rated the realism of SandCanvas as 4.4 on a scale of 1 (not realistic) to 5 (very realistic). However, one user commented that it has yet to achieve the fluidity of real sand. We believe this is because our current implementation does not model sand grain momentum.

Users commented that they liked the look and feel of SandCanvas and indicated that it was very easy to learn: average-rating 4.6 on a scale of 1 (extremely difficult) to 5 (extremely easy). In the initial exploration step, five functions (undo, redo, reset, change texture, and change color) were discovered by all users. All but one user discovered capture frame and all but three guessed the purpose of the record session button. No users discovered how to pour sand or record gestures, but we expected that these bi-manual functions would require training. After the training step, all users understood all features. One user commented, "*After going through the instructions once, the functionalities are quite obvious*".

Q2. Do users find the novel features of SandCanvas useful, and can they apply them effectively?

To better understand the relative merits of our novel features, we recorded the number of times each feature was used in the free task step. Our seven participants spent a total of 159 minutes on free tasks, during which we logged 380 feature usages (average 2.39 features per minute).

Figure 3-13 summarizes feature usage. Each user made moderate use of most features, though undo and change texture stood out, accounting for 81 (21%) and 60 (16%) occurrences of all feature usages logged. Only four users took advantage of gesture playback, but they made heavy use of this feature (74 times or 29% of all feature usage). We also note that no users took advantage of the record session function, because post-production was outside the scope of this study.

The fact that almost all users used most of the features multiple times indicates that users found them useful in creating art works on SandCanvas. We are also encouraged to find out a number of users (3 of 7) embrace the more advanced gesture record functionalities and frequently used it in their art creation process.

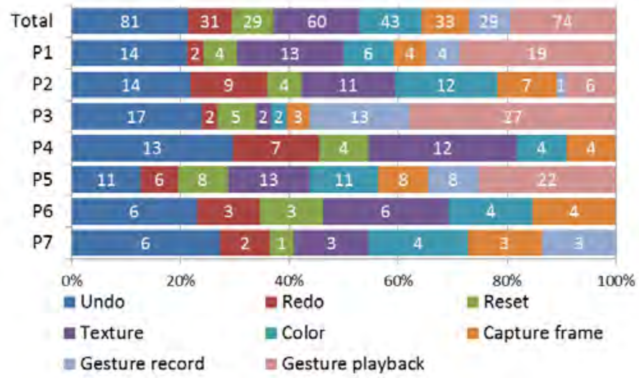


Figure 3-13: Feature usage counts among participants.

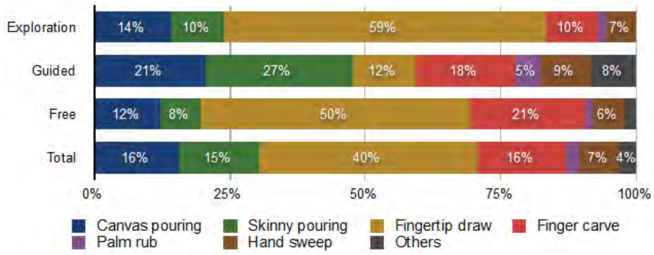


Figure 3-14: Overall usage of techniques in evaluation steps 1 (exploration), 3 (guided task), and 4 (free task).

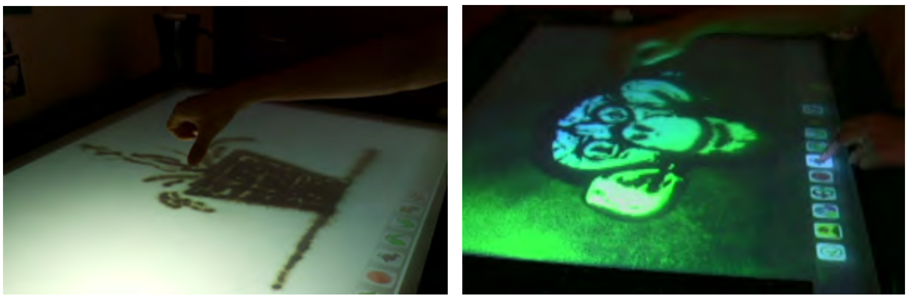


Figure 3-15: Drawing and animating with pouring (*left*) and manipulation (*right*).

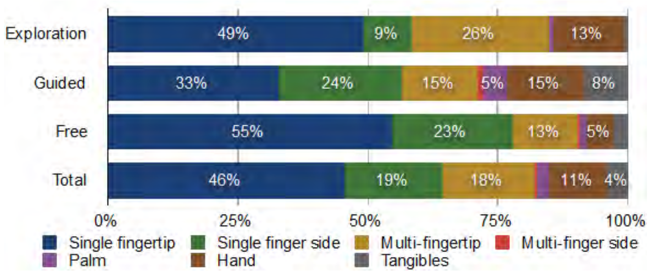


Figure 3-16: Another dimension of gestures we are particularly interested in is the distribution of different parts of hand (or actuation) in creating art works.

Q3. Can we gain further insight into the expressiveness of this new medium by analyzing gestures that users employ?

In addition to our feature analysis, we wanted to enhance our understanding of the unique affordance of SandCanvas by analyzing the gestures employed by users. We analyzed the video tapes of the user evaluation and classified all 3580 gestures they performed in steps 1 (exploration), 3 (guided task), and 4 (free task). Note that step 2 was for training purpose only; therefore it is not included in the analysis. Step 1 is included because it demonstrates users' the initial reaction to SandCanvas, which can be contrasted with later stages when they become more experienced.

Figure 3-14 shows the breakdown of common sand animation techniques used in the three evaluation stages. Overall, pouring accounted for 31% of gestures and manipulating accounted for 69%. The most common technique was fingertip draw (40% of gestures). Finger carve, canvas pour, and skinny pour are the next most popular techniques (each contains 15-16% of all gestures). Hand sweep (7%) and palm rub (2%), were used less frequently, but they did play a role in drawing. Finally, we note that tangibles were also used occasionally (4%).

We also observed two approaches to creating artworks in SandCanvas (**Figure 3-15**), each with a different distribution of drawing techniques. In the free task, five users took a subtractive approach, in which sand is manipulated to create shapes. Two users took an additive approach, in which shapes are made by pouring sand onto the canvas.

Different stages of our experiment also showed different distributions of drawing techniques. The guided task requires uses to take an additive approach, in which shapes are made by pouring sand onto the canvas. In the free task, however, 5 of 7 users took a subtractive approach, in which shapes are created by drawing in sand. Because of this, pouring was used much more frequently in the guided task (48% of gestures) than in free tasks (18% of gestures). However, the beauty of Sand

Animation is that users are free to switch between these approaches, making smooth and seamless transitions to create interesting and often surprising effects, such as **Figure 3-6**.

In addition to high level techniques, we analyzed gestures according to our low-level taxonomy to get a detailed sense of how users employed their hands. Almost all gestures (99%) were dynamic rather than static, which helps to justify our efforts to support dynamic gestures. Precision was more or less evenly split between coarse (42%) and fine (59%), indicating the variety of gestures performed. 7% of gestures were bimanual, and almost all of these were performed by three participants drawing symmetrical shapes. This confirms our intuition that bimanual interaction would be an essential part of this medium.

Figure 3-16 shows the hand actuation dimension of our taxonomy for gestures performed in the exploration, guided task, and free task steps of our experiment. This data shows that SandCanvas truly leverages many parts of the hand. The most common gestures were single fingertip (46%), followed by single finger side, multi-fingertip, and hand gestures (19%, 18%, and 11%, respectively).

Tangible gestures were less common (4%), but played a vital role in the artworks where they appeared. One user used a sheet of paper to pour over a very large area, and another used a sharp object to draw a star shape. Palm gestures were rare (2%) and served the same purpose as hand gestures. We have noted elsewhere, however, that palm gestures are useful for creating cloud-like shapes. Finally, this hand actuation data highlights the importance of capturing the full region of contact between the user's hand and the drawing surface. Single finger side and hand gestures together accounted for 30% of all gestures. None of these gestures could have been captured by a system that reduced users' input to a set of points. However, we did not find any significant pattern differences in gesture profiles between amateur and expert users.

The gesture analysis presented here demonstrates that SandCanvas truly capitalizes on the expressive vocabulary of hand gestures found in sand animation. This was possible only because our implementation captures the full area of contact with the surface, and because it carefully balances sand simulation speed and accuracy. All parts of the hand can be used to produce creative works of art.

Q4. Is there evidence that SandCanvas facilitates creativity?

We found four classes of evidence to support this. The richness of gestures captured by SandCanvas provides some initial evidence that it facilitates creativity. As one user put it, "[The] use of different parts of hand for direct manipulation inspires creativity." This rich input inspired users to envision radically new uses that go beyond performance art. As another user reported, "*I like the instant gratification of it. I would like to use it for brainstorming and story prototyping, because it's so easy and quick to create*." Because these gestures are more intuitive than the complex tools in conventional interfaces, SandCanvas may also reduce memory demands, making it easier for users to enter a state of creative flow [28].

Second, the variety of artworks produced by users during the free task is also evidence that SandCanvas facilitates creativity. In the 159 minutes our 7 users spent on this task, they were able to create 13 different artworks, with 161 distinct drawings (about 1 drawing per minute). Among the 13 artworks, eight of these were performance artworks, four were stop motion animations, and one was a static image. The subject of these artworks ranged from portraits to dynamic landscapes to action sequences with multiple characters. The ability of users to create such a collection of artwork in a short time demonstrates the potential of SandCanvas as a creative medium.

Third, we observed users devising creative strategies for producing similar effects. For example, one training task asked users to draw a snail. To draw the same spiral shape, one used a finger carve gesture, while the other used his whole hand.

Finally, we found that SandCanvas's novel gesture recording feature inspired particularly creative uses. Many users took advantage of gesture recording to clone objects on the canvas, (e.g., to quickly create a crowd of people). One user recorded several drawings of words and played them back all at once to give an impression of many simultaneous speakers. Another user interleaved playback of ring-shaped pouring and drawing gestures that produced a complex interplay between gestures. Finally, one user combined gesture playback with undo to produce stop-motion animation. He recorded the drawing of a spaceship and moved it across the canvas with a sequence of capture frame, undo, and play gesture operations. These unexpected and creative uses of SandCanvas' gesture recording function show that SandCanvas is truly a creative medium that goes beyond traditional sand animation.

The evaluation we have presented here has demonstrated the intuitiveness of SandCanvas and the effectiveness of our approach to modeling real-time interactions between hand and sand. But it does much more. It has also given us a deeper understanding of the affordances of this new medium, particularly the variety of gestures at users' disposal. Finally, it has shown how SandCanvas's modeling approach, novel creative tools, and intuitive UI combine to produce an important new creative medium.

3.7 Summary

The design of SandCanvas started by looking at the materials, medium, artifacts, and process. The elegance of sand animation lies in the seamless flow of expressive hand gestures that cause images to fluidly evolve, surprising and delighting audiences. Our analysis suggested that sand animators use expressive hand gesture to produce a variety of visual effects.

While physical sand animation already possesses these properties, the design of SandCanvas enhances them. Producing this new artistic medium required us develop a new approach to real-time sand simulation that strikes a balance between speed and realism. It also required a simple and intuitive UI that would enable users to employ our new features effectively. SandCanvas's color and texture features enable faster, more dramatic transitions, while its mixed media and gesture recording features make it possible to create entirely new experiences, which goes beyond traditional sand animation. Session recording and frame capture complement these capabilities by simplifying post-production of sand animation performances.

The evaluation of SandCanvas shows that we succeeded. When we analyzed it with respect to common sand animation techniques and our own taxonomy of gestures, we found it to be a genuinely rich artistic medium that enhances both professionals' and novices' opportunities for creative expression. We found that users used different parts of their hands intuitively, and used different (additive and substractive) painting techniques, demonstrating the effectiveness of the design approach.

4. VIGNETTE: A SKETCH-BASED TOOL FOR PEN-AND-INK ILLUSTRATION

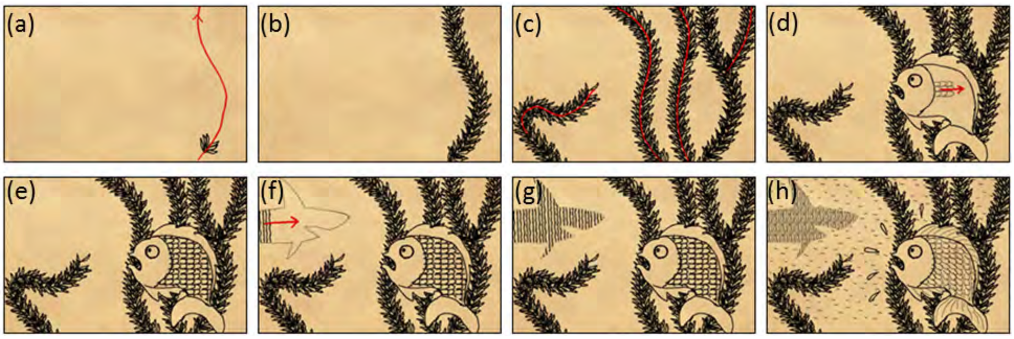


Figure 4-1: The steps of a pen-and-ink illustration with Vignette from scratch (a) Draw leaf strokes (black) and gesture (red) (b) Texture created from gesture and strokes(c) More textures (d) Draw scale strokes and gesture (e) Region filed with scales (f) Draw hatching strokes and gesture (g) Fill region with hatching (h) Final illustration created in minutes.

Chapter 3 presented the first case study, a digital tool inspired by sand animation. This chapter presents the second case study, a sketching tool for pen-and-ink illustration. As an artistic medium, pen-and-ink illustration contrast sand animation in many ways. First, unlike pen illustration, sand animation was a performance art. Second, pen-and-ink illustrations rely on ink as medium and variety of pens as tools, in contrast to sand and human hands in sand art. Finally, as for artifacts, pen illustrations consist of strokes and textures, in contrast to layered sand texture in sand art. Inspite of their differences in medium, materials, tools, techniques and artifacts, we will exhibit a similar design process to design Vignette.

4.1 Background and Motivation

Pen and ink illustration is a popular artistic medium that can be seen in textbooks, repair manuals, advertisements, comics, and many other printed and digital media. Illustrations typically incorporate a wealth of textures, tones and artistic styles. These effects take significant amounts of skill, artistry, and patience to create.

Many research systems [10, 37, 133, 134, 166, 167] can render scenes in the style of pen-and-ink illustrations. Also, professional tools like Illustrator, Photoshop, Comic Studio and InkScape can synthesize customized textures. These tools are powerful and widely used, but they fall short of preserving two key properties of traditional paper-based pen-and-ink illustrations.

The first key property is artists' rich personal style, as seen in **Figure 4-2**(a). Arthur L. Guptill explains, "*... the more conventional the art, the greater the opportunities for originality. We might go so far as to say that there is perhaps no medium offering one a better chance for development of a personal technique than the pen, for pen drawing is akin to handwriting, and just as no two people write alike, so no two people draw alike…*" [48].

Tools developed for pen-and-ink style renderings [10, 37, 133, 134, 166, 167] require some kind of 3D models or 2D images to serve as the template for guiding the generation of textures. As a result, these tools can create high quality pen-and-ink style drawings, but there is little room for variation and artistic styles. Similarly, most tools for 2D texture generation and manipulation lack the natural feel of pen-and-ink drawing (**Figure 4-2**(b)).

The second key property that existing tools fail to preserve is the workflow of pen and ink illustration. Generating a drawing from 3D scenes or images destroys this workflow completely. Texture generation tools do use artists' pen strokes, but much of the creation process with these tools is devoted to parameter tweaking. These tools can produce diverse effects, but they are often difficult to learn, and tedious to apply.

This chapter presents Vignette, an interactive system for pen-and-ink illustrations that uses free-form gestures for texture design and manipulation. Vignette provides tools to design, arrange, and manipulate pen-and-ink illustration textures through simple gestures. Vignette preserves traditional pen-and-ink illustration workflow while accelerating the creation of textures from user defined example strokes. This project focus on drawing from scratch, using textures generated entirely from artists' hand-drawn strokes to preserve the original style and signature of individual artists.

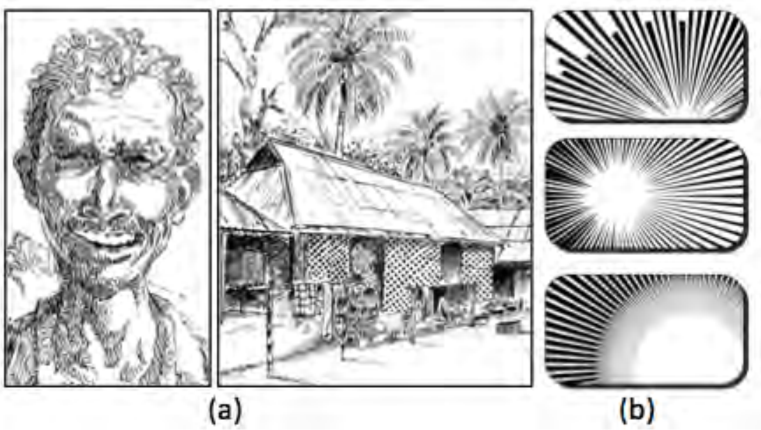


Figure 4-2: a) Illustrations made by hand have distinctive styles b) Illustrations made in comic studio have a mechanical look.

In this system, the user draws a small fragment of the target texture, specifies the type of texture by choosing a tool, and gestures to define the growth of the texture (**Figure 4-1** (a), **Figure 4-1** (d), **Figure 4-1** (f). The system completes the texture, preserving the style of the example strokes (**Figure 4-1**(b), **Figure 4-1**(e), **Figure 4-1**(g)). The user then interactively refines the textures, tones, perspective view, sweep and orientation of the texture to achieve desired results (**Figure 4-1**(h)). Using Vignette, even first-time users can create complex and expressive illustrations within minutes.

This project presents the following contributions:

- An analysis of traditional pen-and-ink illustration workflow and artifacts that guides interface design.
- The Vignette system, which facilitates texture creation while preserving this workflow.
- An evaluation with four artists that shows how Vignette reduces the tedium of texture creation.

After reviewing related work, this chapter presents our analysis of the traditional pen and ink illustration process and categorizes the textures used by artists. We then describe the interface of Vignette, which is based on this analysis, and follow this with Vignette implementation details. Finally, we present an evaluation of our system with 4 professional artists.

4.2 Tools and Techniques for Texture Illustration

The methodology we describe here builds on previous work for pen-and-ink illustration rendering, texture synthesis, and design of digital tools inspired by traditional approaches to creating artifacts. We discuss representative examples of previous work in these areas below.

4.2.1 Pen-and-Ink Rendering Systems

A number of systems render illustrations in a pen-and-ink style. Some are geometry-based [133, 166, 167], taking 3D scene descriptions as input, while others take 2D images as input [37, 134]. The tones and textures in these systems are therefore guided by the underlying 3D geometry or 2D image. Instead, we focus on workflows that allow illustrators to produce artworks from scratch, where no scene model or image exists. Our system analyzes reference patterns and gestures drawn by artists to synthesize new patterns with similar perceptual properties.

4.2.2 Commercial Drawing Applications

Applications like Adobe Photoshop, Adobe Illustrator, Comic Studio, Sketchbook Pro, InkScape, and CorelDRAW have become mainstays of digital artwork creation. In pixel-based applications like Photoshop, duplicating an example patch in multiple layers or using pattern brushes can speed up some repetitive tasks that illustrators encounter [100]. With pixel-based approaches, however, it is hard to control density, add variation, or deform textures.

Vector graphics editors like Adobe Illustrator and Comic Studio are very powerful but awkward for illustration. These tools allow artists to define custom textures that can be controlled by a set of parameters, but the resulting textures tend to lack the subtle variations found in traditional illustration. Furthermore, tweaking the many parameters to get a desired texture is tedious and shifts attention away from the artwork itself. These tools are oriented more toward graphic design than pen-and-ink illustration.

4.2.3 Texture Synthesis

One way to preserve personal artistic style is to create larger textures from user drawn examples. Texture synthesis methods synthesize new textures from texture samples in such a way that, when perceived by a human observer, they appear to be generated by the same underlying process. The idea of synthesizing textures, both for 2D images and 3D surfaces, has been extensively addressed in recent years (see a survey of this type of work in [164]). However, the basic representations in most existing texture synthesis methods such as pixels [60, 163], vertices [156], voxels [84] or parametric descriptors [171] cannot adequately represent individual or discrete elements with semantic meanings. Moreover, subtle variation in the reproduced pattern is desirable for changing density and avoiding regularity. It is difficult to achieve such variation with pixel-based texture synthesis.

The use of vector-based descriptions of an input pattern for synthesis is explored in [9, 10, 47, 52, 60]. These descriptions are more expressive and allow higher-level analysis than pixel-based approaches. However, [9, 10, 52] do not reproduce the interrelation of strokes within a pattern, and are thus limited to hatching and 1D synthesis only. Ijiri et al. [60] presented a method for synthesizing 2D elements by locally growing a 1D ring of elements in a neighborhood around an example. Their examples are points, not strokes, which limits the user to synthesizing dot patterns.

Barla at al. present a synthesis technique [47] that can automatically generate stroke patterns based on a user-specified reference stroke pattern. This is an extension of texture synthesis techniques to vector-based patterns. However, both Barla et al and Ijiri et al use triangulation to perform 2D synthesis. This approach cannot handle elements with complex shapes that are closely correlated with spatial distributions. Instead, we use a data-driven texture optimization method [97] for stroke synthesis.

Vignette provides a novel way to design and manipulate textures for pen-and-ink illustrations completely from scratch. We integrated texture synthesis methods with free-form gestures to provide powerful texture tools that help artists create beautiful artworks.

4.3 Pen and Ink Illustration: Medium

As illustrated earlier, one of the key affordances of pen and ink illustration is its ability to accommodate a wide range of artistic styles. In this section, we review principles of pen-and-ink illustrations and introduce some terminologies.

4.3.1 Strokes

Strokes (**Figure 4-3**(a)) are the building blocks of textures. For centuries of pen and ink illustrations, artists have infused drawings with their signature styles through careful use of individual strokes. Strokes become textures when drawn in groups. (**Figure 4-3**).

4.3.2 Textures

A texture is a collection of strokes that gives an object or scene the illusion of shape, surface properties, and lighting. In a texture, individual strokes are not of critical importance, but collectively they can clearly indicate the difference between textures like smooth glass and old knotted wood (see **Figure 4-4**(a)-**Figure 4-4**(f)).

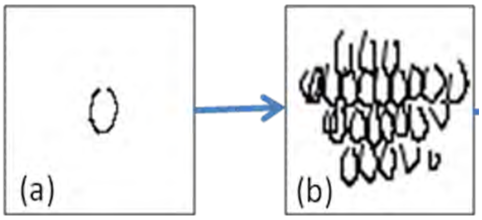


Figure 4-3: (a) Individual strokes (b) combine to form textures.

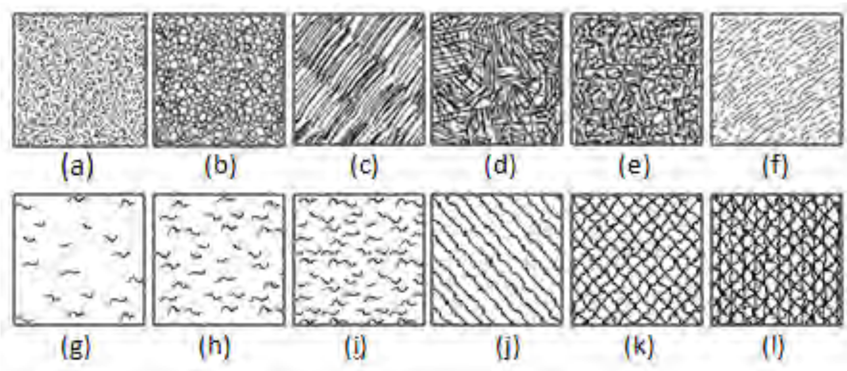


Figure 4-4: (a)-(f) Different kinds of textures. Variation of tones by (g)-(i) changing the density of strokes (j)-(l) subsequent cross-hatching

4.3.3 Tones

Tone (also known as "value" or "density) refers to the density of strokes in a texture. The tone is the ratio of black ink to white paper over a given region of the texture. **Figure 4-4**(g)-**Figure 4-4**(i) and **Figure 4-4**(j)-**Figure 4-4**(l) shows the variation of tones with the same texture to indicate the brightness of a surface.

Together, stroke, texture, and tone provide artists with a rich language for producing expressive illustrations with a variety of personal styles [48].

4.4 Pen-and-Ink Illustration: Artifacts and Process

In this section, we examine the artifacts and process of creating pen-and-ink illustrations. We analyzed the textures in 56 illustrations to identify opportunities for automation. Our findings can be used to guide the design of pen-and-ink illustration systems.

4.4.1 Traditional Illustration Workflow

While the process of pen-and-ink illustration can vary from artist to artist or even between one artists' illustrations, the illustration process usually follows five steps [109]:

Step 1: create outlines of simple geometric shapes and regions of interest with a pencil. The drawing at this step is typically light and erasable.

Step 2: pencil in details and shadows. Iterate until the outline and any object highlights are well-defined.

Step 3: begin filling in the detailed textures, starting with small areas of example texture. We call these small example textures patches.

Step 4: repeatedly apply patches to fill in the outlines.

Step 5: add or modify details to complete the illustration.

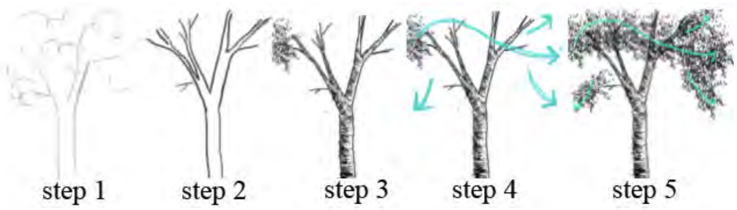


Figure 4-5: The steps in traditional pen-and-ink illustration.

Steps 1 and 2 determine the high-level structure of the illustration with shape outlines, their spatial layout, and indicators of properties such as shadows and highlights. Step 3 determines the detailed textures and tones of the shapes or regions determined in step 1 and 2. Essentially, these three steps contain most of the essential elements to uniquely define the style and content of an illustration.

In step 4, the artist repeatedly applies the various textures and tones to all shapes and regions in the illustration. This fourth step is the most tedious and contributes the least to the uniqueness of a pen-and-ink illustration. However, the illustration cannot be completed without it.

In the last step, the artist touches up the illustration with final details and adjustments. Note also that artists often iterate this process and jump between steps.

Analyzing this workflow, we found that a major component of an artist's personal drawing style lies in her procedure (or muscle) memory of using the pen, which is reflected in her strokes [109]. Using a third party image or model cannot preserve this unique style. If the artist were asked to produce the texture separately, saving it into an image before applying it to the drawing, it would break the creative flow of the drawing process. Therefore, we believe it is important to allow the artist to define both the outline and example textures from scratch using her hands.

Finally, we note that step 4 in the traditional workflow is the most repetitive and time consuming. Consequently, it is quite suitable for automation.

4.4.2 Texture Automation Techniques

To inform the design of systems that automate step-4 of the traditional illustration workflow, we examined the kinds of textures that professional pen and ink illustrators use. We analyzed 56 rich pen and ink illustrations by 32 artists, mostly taken from The Technical Pen [145] and Rendering with Pen and Ink [48]. After analyzing the textures in these illustrations, we classified them according to techniques artists could use to automate the filling-in process. We identified three techniques: brushing, flood filling, and continuous hatching.

As we explained in Related Work, brushing and flood filling techniques exist in current graphical tools, but they are tedious, awkward, and do not preserve artists' style. Continuous hatching cannot be found in these tools at all. Vignette provides all three techniques, and it uses texture synthesis of vector geometry to produce pleasing results that preserve artists' style.

It should be noted that these techniques cannot reproduce all textures effectively. Automation requires textures to be repetitive so that a computer can synthesize them from example patches. Some textures have so much variation that they cannot be synthesized from patches.

In the following paragraphs, we describe these texture filling techniques along with applications and variations.

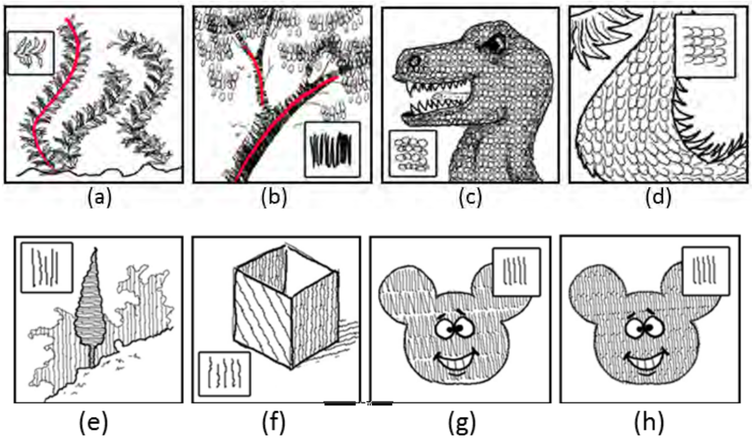


Figure 4-6: Applications of texture automation techniques. (a-b) Brushing. (c,d) Flood Filling. (e-f) Continuous Hatching. (g) Flood fill from the example patch (inset). (h) Continuous hatching from the same patch (inset), in which discrete strokes are uniformly stitched together.

Flood Filling

A small set of discrete strokes can often be used to fill up a region. Flood Filling can also be done in a particular orientation to follow the contour of the volume or shape (**Figure 4-6**(d)). We identified this effect in 25 out of 56 illustrations. Applications of flood fill include stippling (where tone and textures are applied with small dots and strokes (**Figure 4-4**(a)), clothes textures, walls, illustrations, wood, landscape etc.

Brushing

In these textures strokes are augmented along a line, rather than filling up a 2D region (**Figure 4-6**(a), **Figure 4-6**(b)). In our analysis, this type of synthesis was more common than the other two (37 out of 56 illustrations). These textures are commonly applied to create a wide range of effects including hatched (**Figure 4-4**(j)) and cross-hatched lines (**Figure 4-4**(k & l)), landscape drawings for trees and grasses (**Figure 4-6**(b), and many other complex textures.

Continuous Hatching

Continuous hatching is a set of closely spaced parallel lines from one edge of a shape to another, similar to symbolic indication of a cross section in an engineering blueprint (**Figure 4-6**(f)) creating tonal and shading effect. Continuous hatching is different from brushed hatching, because the synthesis is two dimensional, i.e. it fills up a 2D region instead of extending along a 1D line. However, unlike the discrete elements in flood-fill, the lines are connected with each other to create longer lines and fill up a region (**Figure 4-6**(e), **Figure 4-6**(f)). Prevalent application of continuous lines includes architectural drawings, cross-hatching, and portraying the illusion of depth (**Figure 4-6**(e)).

4.5 Vignette: Interface and Interaction

Our analysis of traditional pen-and-ink illustration processes and artifacts helped us to build Vignette, a texture synthesis system that is based on the traditional illustration workflow. Here we present Vignette's user interface. We begin with Vignette's toolbars and palettes, then describe our workflow, and close with interactive refinement tools.

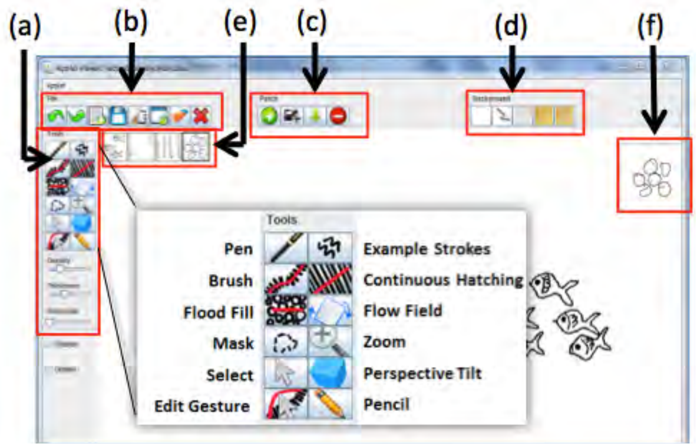


Figure 4-7: The User Interface of Vignette (a) Drawing and texturing tool (b) File/edit toolbar (c) Patch toolbar (d) Background palette (e) Patch palette (f) Selected patch

4.5.1 Vignette's Toolbar and Pallettes

As shown in **Figure 4-7**, Vignette has four toolbars located around a central drawing canvas. These are the main toolbar (**Figure 4-7**(a)) for drawing and texturing tools; a file/edit toolbar (**Figure 4-7**(b)) for common commands; patch toolbar (**Figure 4-7**(c)) for adding, updating, importing and deleting patches; and a background palette (**Figure 4-7**(d)).

The top left region of the drawing canvas is reserved for a palette of patches (**Figure 4-7**(e)); each patch is shown horizontally from left to right in small rectangles according to its creation order. Patches are example texture patterns created using the Example Strokes tool (details later). There is also a larger rectangle on right that displays the currently selected patch (**Figure 4-7**(f)). The remaining area of the drawing canvas is for freeform pen-and-ink illustrations.

Vignette's main toolbar supports the five steps of the traditional illustration workflow. It has 12 buttons and 3 widgets (**Figure 4-7**(g)), which can be grouped into 5 categories:

1) Tools for outlining. This category has the Pencil tool. In traditional pen-and-ink illustration, a pencil is used in steps 1 and 2 to outline the high level structure of an illustration. Similarly, strokes drawn with Vignette's Pencil tool are stored on a separate layer which can be easily removed after finishing the illustration.

2) Tools for detailed drawing. This category has the Pen tool, which is used to draw detailed non-repetitive strokes and fine details of an illustration, such as a person's eye.

3) Tools for specifying example textures. We created the Example Strokes tool to support the third step in the traditional workflow. Strokes drawn with this tool are collected into patches and later applied to different regions.

4) Tools for growing textures and tones. Tools in this category support the fourth step in the traditional workflow: Mask, Brush, Continuous Hatching, and Flood Fill. Mask

defines a closed region to be filled up with the target texture. The others will be covered in more detail later.

5) Tools for texture layout and refinement. Textures can be refined interactively using the Flow Field, Perspective Tilt, and Edit Gesture tool (explained later). In addition, a number of slider widgets can be used to adjust the tone, stroke width, grayscale value of the textures.

4.5.2 Workflow In Vignette

The following steps illustrate the typical drawing workflow in Vignette:

Step-1: Users can draw a rough outline of the illustration using the Pencil tool.

Step 2: After the high level structure is defined, users can select the Pen tool to draw the detailed outlines. Users can use the Mask tool to define a region to be filled with texture.

Step 3: The user can then draw part of the texture using the Example Strokes tool. (**Figure 4-8**(a), **Figure 4-9**(a), **Figure 4-10**(a)).

Step 4: The user then selects a texture filling tool (Brush, Continuous hatching, or Flood fill) and gestures to specify how the texture should be filled in (**Figure 4-8**(a), **Figure 4-9**(a) and **Figure 4-10**(a)). The example strokes are automatically collected into a patch, while the direction and curvature of the gesture specify the reference orientation of this patch. The system then generates the rest of the texture from the example patch to fill up the region (**Figure 4-8**(b), **Figure 4-9**(b), **Figure 4-10**(b)). To understand how Vignette collects strokes into patches or fills in textures, refer to Generating Patches From Example Strokes and Texture Synthesis in our Implementation section.

Step 5: After generating the textures, users can interactively manipulate and fine-tune the textures to achieve the desired artistic effects, as explained in the following section.

4.5.3 Interactive Refinement in Vignette

Vignette's aids creative exploration by providing high-level controls for manipulating textures. Here, we briefly describe our interactive refinement capabilities.

Editing Textures

To edit a texture, it must first be selected using the Select tool in the main toolbar. The corresponding patch appears as current patch in the top right of the canvas. As the user edits the example patch, the system interactively changes the selected output texture to reflect the change in the example patch (**Figure 4-10**(d)).

Editing Tones

Users can edit the tone or density of a texture by manipulating a slider. Since each of the elements is represented by single point in the texture, we simply scale the density of the positions of the elements and re-render the elements. Variation of textures by tone editing is illustrated in Figures **Figure 4-8**-**Figure 4-11**.

Editing the sweep of a texture

Users can interactively edit the sweep of a texture by editing the curvature of the reference gesture [57] (**Figure 4-8**(c)).

Texture Flow

By default, textures are filled in uniformly as if on a flat surface. Often, however, users may wish for textures to gradually change as they fill a region. Vignette provides two tools for this: perspective tilting and flow fields.

Perspective tilting is a technique for depicting 3D surfaces in illustrations. In perspective drawing, objects are drawn smaller (or "foreshortened") as their distance from the eye is increased. In our system, users can manipulate the perspective view of a texture without any underlying 3D information by manipulating the eye position with gesture with respect to the texture. Currently, our system supports one point perspective tilting (**Figure 4-10**). To tilt a texture, the user selects the Perspective Tilt tool in the main toolbar and drags the pen. The angle between the first point and the current point determines the direction of foreshortening, and the length determines the amount of foreshortening.

Flow fields allow users to specify the direction of the texture as it flows across a surface. In Vignette, users can select a texture, and then use the Flow Field tool in the main toolbar to adjust the direction of this field. Gesturing with the Flow Field tool tilts the field in the direction of the gesture, which orients the texture's strokes along the gesture. This is shown in **Figure 4-11**.

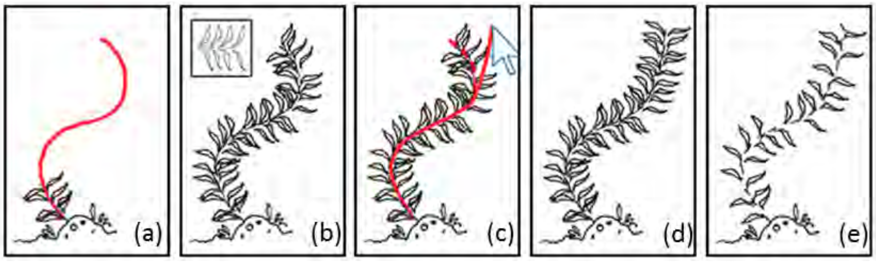


Figure 4-8: Brush operation and editing the curve of a texture. (a) Example strokes and user gesture (red). (b) Brush tool generates a texture from the example patch (inset). (c) User selects the curve editing tool and drags the mouse to sweep the curve. (d) New texture after the editing. (e) Tone variation (decreased density).

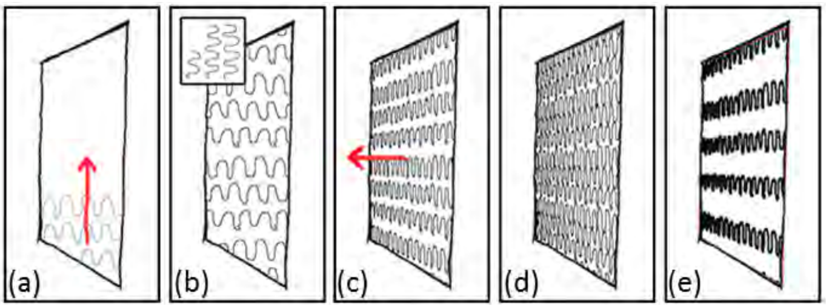


Figure 4-9: Continuous hatching and perspective tilting of a texture. (a) Example strokes and user gesture. (b) Continuous hatching creates a texture from the example patch. (c) User selects the tilt tool and creates a perspective tilt by dragging the mouse left. (d) Tone variation. (e) Tone and stroke width variation.

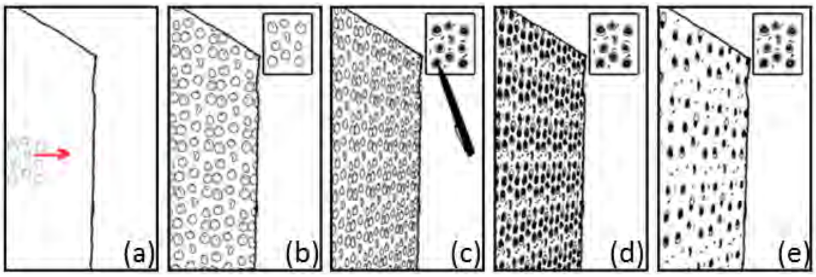


Figure 4-10: Flood fill and texture editing. (a) Example strokes and user gesture. (b) The flood fill creates the textures. (c) Perspective tilting and editing the source patch (inset). (d) The texture is updated interactively. (e) Tone variation.

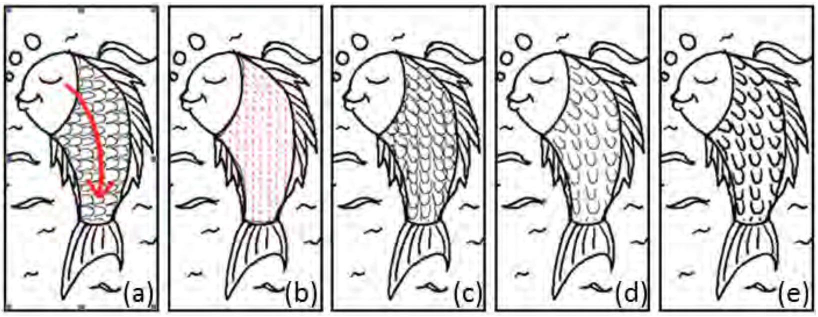


Figure 4-11: Orienting the elements with interactive flow field. (a) User gesture. (b) Underlying vector field from user gesture. (c) Rendering the elements along the vector field. (d) Tone variation. (e) Variation of stroke width of the strokes.

4.6 Implementation

In this section we discuss how Vignette supports the texture synthesis techniques described in previous sections.

4.6.1 Generating Example from Example Strokes

The first step of our method is to generate a patch from the example strokes near a user gesture. After the user draws example strokes (black strokes in **Figure 4-8**(a), **Figure 4-9**(a), **Figure 4-10**(a)) and gestures over them (red curve in **Figure 4-8**(a), **Figure 4-9**(a), **Figure 4-10**(a)), example strokes near the gesture are gathered into a patch. The system clusters strokes together into elements by merging the strokes with overlapping bounds. Our intent was for an element to be a cluster of strokes that is perceived as a single feature by the user (as in [47]).

4.6.2 Texture Synthesis

The example patch provides a higher-level, perceptually meaningful description of example elements. The next step is to create a larger texture by synthesis from the example patch.

Each of the individual strokes is represented with a set of 2D points. An element is a group of strokes. In the textures, we represent each element by a point sample, which is the centroid of the element. During synthesis, we compute only the sample point without considering any other information of the original elements, like their geometry and appearance. After synthesis, we replace the sample points with the output elements.

Now we will briefly describe the synthesis techniques of the three tools: *brush*, *continuous hatching*, and *flood fill*.

Brush

The brush produces a 1D synthesis of elements [47] along the user gesture. Once the patch is computed, the elements are appended interactively along the gesture. The distance between consecutive elements is computed from the example patch. The orientation of each element is computed from the tangent of the corresponding point on the gesture.

Continuous Hatching

Continuous hatching synthesis is performed in three steps. First, we generate the example patch and use the gesture direction (**Figure 4-12**(a)) to perform a 1D synthesis along the gesture direction (**Figure 4-12**(b)). Second, we duplicate and paste each element to fill up the region on either side of the gesture (**Figure 4-12**(c)). Finally, we connect and merge the elements across the vertical direction using a simplified version of stitching [40] to create long seamless elements (**Figure 4-12**(d)).

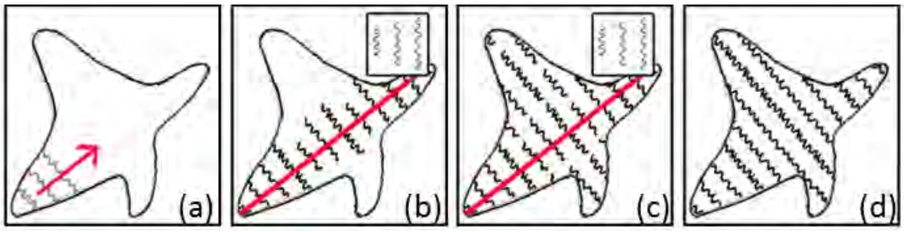


Figure 4-12: The steps for continuous hatching

Flood Fill

For flood fill, given an input exemplar patch ***I*** and an output masked area, the goal is to synthesize an output texture ***O*** that contains elements similar to exemplar patch ***I*** (**Figure 4-14**).

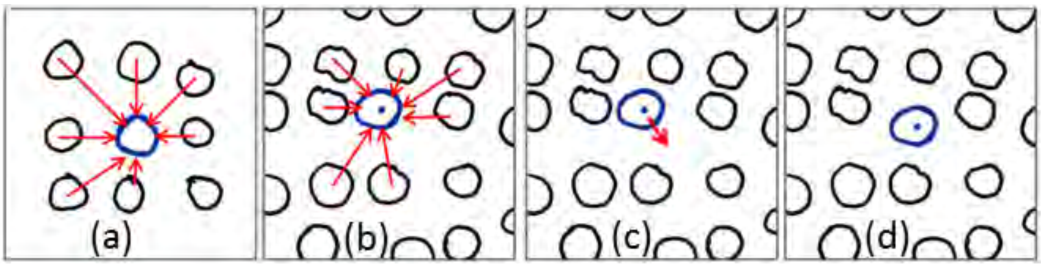


Figure 4-13: (a) The input example patch. (b) The output texture. In the *search step* for output element s_0 (marked blue in (b)), the algorithm finds the s_i in the input patch (marked blue in (a)) with most similar neighborhood. (c) The *assignment step* computes the new position of the output element that minimizes the energy between corresponding input elements. (d) Finally, the element is moved into its new position.

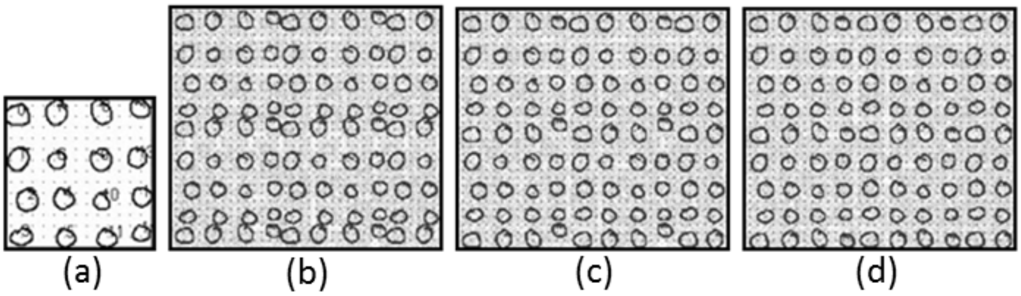


Figure 4-14: The iterative progression of texture optimization. (a) Input patch (b) Output texture after initialization (c) Output texture after iteration 2 (d) Output texture after iteration 4.

In Vignette, we follow the EM methodology in [48] for texture synthesis because of its high quality and generality. This method iteratively places and then adjusts element positions in the texture to minimize the objective function ***E***. The objective function ***E*** is an evaluation criterion that quantitatively evaluates the arrangement of elements with respect to input example patch and performs heuristically chosen tests to try to reduce the energy. The basic solver gradually improves the neighborhood similarity term by iterating the two steps: search and assignment (explained below). The solver gradually decreases ***E*** while improving output quality iteratively (**Figure 4-14**), and continues until the energy function E of output texture ***O*** is optimized.

Initialization: First, we copy the input patch into the output exemplar similarly to tiling methods, but with some random variations in positions.

Search step: During the search step, for each output sample so, we find the input sample s_i with the most similar neighborhood, i.e. minimizing the energy value in accordance to a neighborhood similarity metric (**Figure 4-13**(b)). This search is conducted by exhaustively examining every input sample for each output element.

Assignment step: After computing the best matching input patch, this step computes the position of output elements that minimizes the energy function (**Figure 4-13**(c)) and moves the element to their new positions (**Figure 4-13**(d)).

4.6.3 Interactive Refinement

Flow field: The task of arranging elements according to a gesture can be divided into two sub-tasks: 1) creating an orientation field over the surface from a user input gesture, and 2) rendering elements according to the orientation field.

We have used a vector field to represent the orientation. In our system, user gestures determine the direction of this field at points within a pre-defined distance of the gesture. **Figure 4-11**(b) shows a small number of red vectors that have been set by the gesture in Figure **Figure 4-11**(a). With an orientation field in hand, we then orient the elements in accordance to the vector field using property layers similar to modeling with rendering primitives [141].

4.7 User Evaluation

Vignette has a unique approach to design and manipulation of textures in pen-and-ink illustrations. It keeps the essential steps of the traditional pen-and-ink workflow while providing gesture controls for texture synthesis. There are few existing research or professional tools designed for the same purpose, and none are directly comparable. Adobe Illustrator may be the closest match in terms of texture creation and manipulation, but it is a general purpose graphical editing tool with an entirely different workflow and interaction style. It also has many additional features/functions way beyond the need of pen-and-ink illustration.

Nevertheless, it is important to understand how professional artists feel about Vignette, and how it compares with the traditional pen-and-ink drawing experience and with existing digital tools such as Adobe Illustrator. To do this, we invited four professional artists to use Vignette, while we sought to answer the following three questions.

1. *How do artists generally feel about Vignette? Does Vignette fit their needs?*

2. How does the pen-and-ink illustration workflow in Vignette compare with paper and with digital tools?

3. How are Vignette's features used and accepted by artists? Are there opportunities for improvement?

4.7.1 Participants and Environment

Four professional artists (P1-P4, 3 males, 1 female, age range 23-55 years old) participated in our evaluation. P1 and P2 are accomplished expert artists. They both work as pen-and-ink illustrators, animators, and directors with 15 or more years of experience. P3 and P4 are intermediate level artists trained in design and illustration at universities. Both have 4 or more years of experience in digital painting. All participants are proficient with Flash, Photoshop, Illustrator and many other tools with 4 or more years of experience.

All evaluation sessions took place in a laboratory. Vignette is built with Java and runs on a standard laptop. All drawings were done on a Cintiq 12wx tablet.

4.7.2 Method

The evaluation was conducted in the following three steps.

Training (15-20 minutes): Participants were first given a brief introduction to Vignette. They then received a tutorial, which consisted of a printed sheet with seven practice drawings chosen to demonstrate the interface and features of the system. Participants were asked to create and interactively refine these drawings to achieve the target result. The facilitator did not intervene unless a participant had trouble creating a drawing.

Illustration (40-65 minutes): In this step, participants were asked to create pen-and-ink illustrations. Some of these can be seen in **Figure 4-15** and **Figure 4-16** (far left).

Feedback. (10-15 minutes): Finally, participants answered a questionnaire about Vignette.

We sought to answer our three questions primarily by observing participants and recording their spontaneous comments. The following sections summarize our findings.

4.7.3 Overall Impression

Participants' overall reactions were very positive. During the course of the evaluation, the participants created many illustrations with a wide variety of textures, such as textures for architectural drawings, landscapes, animals, crowds, fireworks, and abstract scenes. Participants responded that Vignette was fairly easy to learn, all giving it a 4 on a scale of 1 (extremely difficult to learn) to 5 (extremely easy to learn). Participants also expressed satisfaction with their artworks, with an average rating of 4.25 on a scale of 1 (extremely unsatisfied) to 5 (extremely satisfied). All commented that Vignette provides a pleasant drawing experience.

Participants found Vignette particularly suitable for two purposes: 1) creating original pen-and-ink illustrations from scratch 2) quickly exploring and experimentation with different types of textures. As mentioned by P1, "*I can draw really quickly, and do a lot of explorations… inspire me to explore more...*"

Participants particularly liked the ability of Vignette to preserve their natural drawing styles. Three participants specifically like the natural and hand drawn scribbling effect of the final artworks, as mentioned by P1, "*it looks like I drew each and every stroke manually… and it is not obvious that the textures were created using a computer tool*".

4.7.4 Workflow and Experience Comparison with Alternatives

Participants were able to create artworks with rich textures in a short time (11 minutes per drawing in average) after only 15 minutes of training (see the artwork in **Figure 4-16**, all except the second artwork were created by the participants during the course of evaluation). All participants commented that it would be very tedious to produce drawings with similar quality either in traditional pen-and-ink style or using another digital tool.

Participants also commented that the advanced digital features make the illustration process enjoyable, which is traditionally very tedious to do. According to them, Vignette is both effective and convenient, and preferable to manual illustrations and other professional tools for pen and ink illustrations. The ability of users to create such a collection of artwork in a short time demonstrates the expressiveness and ease of use of Vignette.

Vignette Vs. Traditional Pen-and-Ink Illustration

Participants commented that although Vignette has many advanced digital capabilities, but the fact that it is designed to follow the traditional pen-and-ink workflow makes the system feels natural to work with and easy to learn and use. On the other hand, using Vignette significantly improved the productivity of drawing.

During the course of evaluation: for example, in Figure 4-15, the user drew a patch of three persons and later used subsequent brush tools for creating a crowd from the example patch. Similar approach was used to draw fireworks. This process follows the traditional pen-and-ink workflow, but is much accelerated. This illustration was created (**Figure 4-15**(right)) in less than 5 minutes, from three example strokes (marked blue) and few gestures (marked red) (**Figure 4-15**(left)).

However, the participants also mentioned that there are certain desirable properties of the traditional pen-and-ink illustrations currently lacking in Vignette, such as the variations of strokes produced by different types of pencils, pens and brushes.

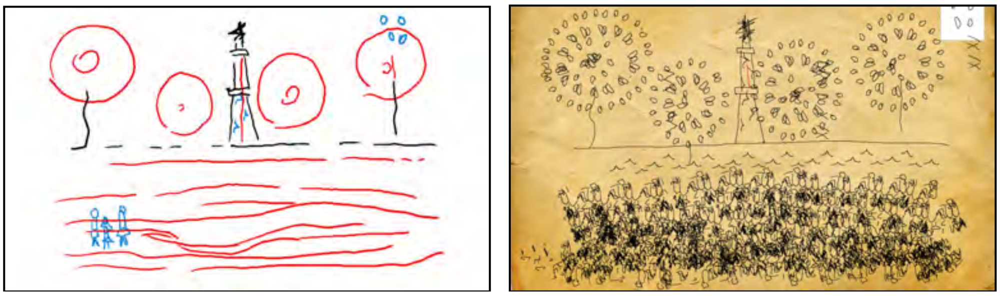


Figure 4-15: The example strokes (blue) and gestures (red) drawn by a participant to produce an illustration in 5 minutes (left). The final illustration (right)

Figure 4-16: Different artworks with Vignette. The artworks took 16, 14, and 19 minutes respectively, completely from scratch

Vignette Vs. Illustrator

According to the participants, Adobe Illustrator is the closest tool they can think of to create and manipulate vector graphics textures. All the participants mentioned that one major difference between Vignette and Illustrator lies in the interface and interaction style. The design of Vignette allows creating illustrations quickly and easily. As mentioned by P2: "*I like the free-form gesture based interaction… it is easy to learn and use… With gestures, a few scratches in the canvas can create illustrations within a minute*".

Compared to Vignette, although Illustrator has many built-in support features for texture and patterns (such as Pattern Brush), it is not optimized for pen-and-ink illustrations. According to P1 – "*Traditional tools have too many functions and options. It is difficult for me to use, (these features) are very often distracting for performing a certain painting*".

Furthermore, participants noticed that Vignette provides additional useful capabilities not available in Illustrator. For example, Illustrator does not provide support for continuous hatching quickly and easily like Vignette. The flood fill effect of Illustrator is simple tiling, hence it can produce results like **Figure 4-14**(b). But, iterative texture optimization reduces the energy function and produce visually better results (**Figure 4-14**(d)), which is suitable for hand drawn textures, since hand drawn textures are not directly tillable most of the times.

4.7.5 Feature Usage and Feedback

To understand the relative usage of features, we recorded the number of times each feature was used in the free task. The four participants spent a total of 66 minutes on free tasks, during which we logged 215 feature usages (average 3.16 features per minute).

Each user made moderate use of most features, though the use of brush to create textures stood out, accounting for 123 (55.4%) occurrences of all feature usages logged. Two of the users made heavy use of brush (71% and 65% of total feature usage respectively). One of these two users didn't use continuous hatching at all, while the other two users made heavy use of continuous hatching (46% and 37% of feature usage respectively). Two users used flood fill.

Almost all participants used all the texture synthesis operations multiple times that indicate that users found them useful in creating art works on Vignette. The participants also like the interactive refinement capabilities, including - flow field, editing tones and textures. According to P1 and P2, editing the flow of elements is very useful and cannot be done easily with other traditional tools.

Opportunities for Improvement

Vignette worked well overall, but we saw several ways to improve it. For certain kinds of textures, having long overlapping example strokes, the synthesis results sometimes look repetitive. Also, like any stroke-based rendering system, Vignette's performance degrades as the number of strokes increases. Our experience is that performance degrades with more than 1000 strokes.

Another limitation of Vignette is that leftward brush gestures appear to create different textures from rightward gestures, because leftward gestures vertically flip the texture. This happened to P1 and P3 a total of 9 times. One user suggested having a preview panel for testing the gesture effects before applying them in final drawings.

4.8 Summary

Like SandCanvas in Chapter 3, the design of Vignette started by looking at the key affordances of the medium, materials, artifacts and process. Our initial analysis, based on descriptive pen-and-ink illustration books re-emphasized the importance of the ability to accommodate the variety of artistic styles. We analyzed the traditional illustration workflow and illustration artifacts to guide designers of illustration systems that preserve this traditional feel. This points out to different texture filling techniques and the tedious steps of the traditional workflow.

Vignette was designed based on these observations. Our exploration of natural workflow and gesture-based interaction was inspired by a traditional approach to creating illustrations. Texture illustration is tedious, but current texture synthesis tools cannot easily capture illustrators' personal style. Furthermore, these tools disrupt the traditional illustration workflow, because they are tedious and draw attention to dialog boxes and away from the illustration itself. Vignette speeds up texture creation while preserving the traditional workflow capturing artists' personal style. The end-user programming approach reduced the tedium, yet preserved the personal and artistic style.

The evaluation presented in this chapter shows how artists can use it to quickly create artworks in their own personal style. The resulting artifacts by the users had hand-drawn organic look and feel, which is desirable to most of the illustrators. In contrast to previous pen-and-ink illustration tools developed by the researchers, our tool emphasizes on style preservation and expressiveness, resulting a novel and unique workflow for pen-and-ink illustrations.

5. DRACO: BRINGING LIFE TO ILLUSTRATIONS WITH KINETIC TEXTURES

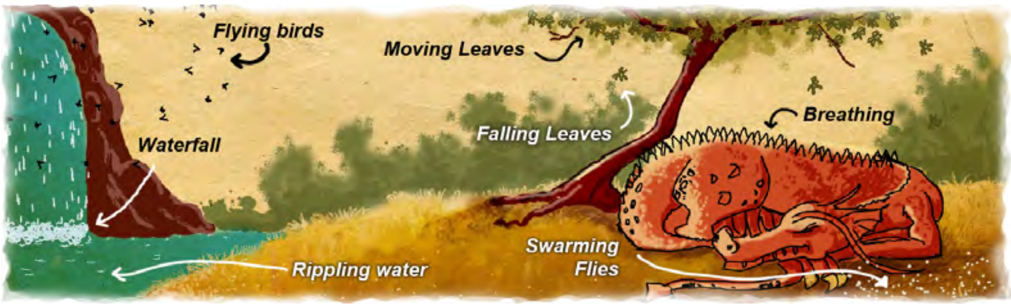


Figure 5-1: A dynamic illustration authored with Draco, capturing the living qualities of a moment with continuous dynamic phenomena, yet exhibiting the unique timeless nature of a still picture.

Chapter 3 presented Vignette, an example of powerful end-user-programming approach for texture illustration, where users sketch a fraction of texture and progression line, and the system performs spatial texture synthesis along the progression line to complete the texture. This chapter looks to extend the spatial synthesis of textures into temporal dimensions, by facilitating the creation of textures with spatio-temporal synthesis. However, there is no physical art medium with such capabilities. In this particular case study, we will take account into existing computer animation tools and techniques to understand the domain, design insights and workflow.

5.1 Background and Motivation

For centuries, people have attempted to capture the living qualities of surrounding phenomena in drawings. Sketching, in particular, is a popular art medium that has also been widely adopted as a powerful tool for communication, visual thinking and rapid design, due to its minimalistic yet greatly expressive nature [18]. While sketches do afford many techniques to convey dynamic motion of objects, such as speed lines [99], arrows [50] or afterglow effect [13], they are inherently static. The goal of this project is to enable artists and casual users alike to enrich static illustrations with intricate and continuous animation effects, while preserving the unique timeless nature of still illustrations (**Figure 5-1**).

In recent years, researchers have developed new tools and techniques for casual animation authoring using sketching [34, 103, 154] and direct manipulation [57, 136]. Such tools typically support basic animations, where motions are defined for individual objects, and then coordinated using a global timeline. In contrast, many natural phenomena are characterized by the coordinated motion of large collections of similar elements, like snowflakes falling to the ground, water drops dripping out of a fountain, or school of swimming fish (**Figure 5-2**). Animating large collections of objects with flexible control is still tedious and cumbersome with existing sketch-based animation tools.

For authoring the animations of object collections, complex software and workflows are often required. Graphic researchers have developed content-specific tools [121] and models [106, 125] for particular phenomena, but these methods are highly specialized and geared towards physical accuracy for professional animators. Furthermore, defining and controlling these behaviors typically require indirect controls, including numerous parameters tweaking and scripting, which makes it difficult to rapidly prototype and experiment with motion effects, even for an expert user. From an interface design perspective, the key challenge to this problem is to formulate a general framework for workflow and controls that is easy to use, but expressive enough to author a wide range of dynamic phenomena.

In this project, we make the following contributions.

- We address this problem by contributing a general framework built around kinetic textures, a novel coherent data structure that consists of a set of similar objects, to which dynamics is applied at the collective and individual scales.

- Built upon this framework, we present Draco, a flexible and fluid sketch-based interface that allows users to easily augment still illustrations with subtle animations of object collections, seemingly bringing to life the moment they portray (**Figure 5-1**). In contrast to traditional animation tools, where animations start and end within a global timeline, our system supports continuous motions, to enrich illustrations with dynamic effects similar in spirit to seamlessly looping video clips [24, 75, 140].

- After describing our framework and authoring system, we report on a user study, conducted with professional animators and casual artists, that evaluates the usability of our system, and demonstrates the variety of animations, applications and creative possibilities our tool provides.

Figure 5-2: Examples of coordinated motion of collections of objects (both natural and artistic). From left to right: snowfall, tree leaves blowing in the wind and falling to the ground, water dripping from a fountain, school of fish, and air blowing off from a fan.

5.2 State-of-the-art Tools and Techniques for Animation

This section reviews prior work in the physical simulation of collections of objects, existing animation tools, and techniques aiming at adding motion to static pictures.

5.2.1 Physical Simulation

Physical simulations of behaviors like that of crowds [106], traffic [142] or flocks [125], excel at creating realistic motion. As such, they have been widely adopted in computer animation industry to create the best dynamic illusion of particular phenomena [22]. Simulating the behavior of these specific collections is mostly geared towards producing very specialized, polished and physically accurate final outcomes. As a result, most simulations do not apply beyond their target phenomenon and require significant expertise to understand the underlying models and parameters. Previous work also suggests that non-physics-based effects are often preferred for their flexibility [22].

Ma et al. [98] recently proposed to generalize physical simulations using a data-driven approach. Their technique consists of injecting the model with granular motion computed from a set of sample animations, combined with global constraints as defined by the user. Our approach draws inspiration from this work in terms of formulating multi-scale motion controls in our framework. In contrast to previous work, we rely on more direct controls, by allowing the user to define the behavior and appearance of groups of objects at the global and local scales through sketching.

5.2.2 Professional Tools

Applications like 3D Studio Max [5], Maya [6] and Lightwave 3D [108] are some of the mainstays of 3D digital animation tools. While these tools allow artists to produce a variety of effects using underlying physics models, they are targeted towards professional animators, and require many parameter tweaking, scripting and domain expertise. Among 2D animation tools, Flash [4] and After Effects particles plug-in [121] are popular. These equally require expertise in scripting and parameter tweaking to animate collections of objects. We propose a system that capitalizes the freeform nature of sketching and direct manipulation to specify and control these types of behaviors.

5.2.3 Adding Motion to Static Pictures

Artists and researchers have explored augmenting images with motion as a way to capture the ambient dynamics of a moment. Video textures [140] provide an infinitely varying stream of images from a video source, preserving the timeless nature of a photograph. Chuang et al. [23] used a semi-automatic approach to animate user-defined segments of a static picture with subtle motion of passive elements in response to natural forces. Inspired by Cinemagraphs [24], Cliplets [75] enable the creation of a visual media that juxtaposes still images with video segments. We bear similar motivation to these works. However, these techniques operate on raster graphics and rely on video sources for animation, providing no authoring capabilities for the motion dynamics. In Draco, users can author and control a variety of dynamic effects completely from scratch with freeform sketching and direct manipulation.

5.3 Analysis: Artifacts and Process

To guide our designs and better understand existing practices for creating coordinated animations of large collections of objects, we conducted a design study. This allowed us to better understand the vocabulary of motion effects and the workflows currently being used today.

5.3.1 Methodology

We used a mixed-method approach for our study, consisting of an analysis of online instructional videos, and a set of interviews with professional animators.

We first collected and analyzed a set of YouTube tutorials for state-of-the-art animation systems including Flash, Maya, 3D Studio Max and After Effects. For each tool, we collected at least one tutorial explaining how to create each of the following effects that involve the animation of collections of objects: rain or snow, falling leaves, swaying grass, flocks or swarms, crowds and water ripples.

To gain further insights, and provide validation of our findings from this analysis, we also conducted interviews with two professional animators. We prompted the experts with scenarios similar to those in the videos that we analyzed, and asked them to demonstrate how they would achieve these effects using their usual tool (both use Maya).

5.3.2 Observed Artifacts: Animation Types

Consistent with prior literature [113], we identified three types of animations used to reproduce effects that involve coordinated motion of collections of objects (**Figure 5-2**): particles systems, flocking and stochastic motion.

Particles Systems [122] are used to model phenomena such as fire, clouds, and rainfalls. Such systems model a collection of dynamic objects (particles) whose behavior is dictated by external forces. Creating particles systems usually requires fine-tuning numerous parameters via indirect controls in a complex interface (**Figure 5-3**). Particles systems are widely used by advanced animators, but their complexity make them poor candidates for casual tools such as Flash, as they require scripting (**Figure 5-3**).

Flocking [125] can be characterized by collections of objects (agents) that exhibit some intelligence in how they interact with their neighbors or environment. The group behaviors can range from unstructured omnidirectional movements (e.g. a swarm of insects), to organized, coordinated motion (i.e. a school of fish). Some professional tools include specialized plug-ins for specific simulations, each coming with its own interface for manually controlling parameters.

Stochastic motion [23] is characterized by the passive movements of elements, such as grass blades or tree leaves, under the influence of natural forces, such as wind. Unlike particles systems and flocking, where objects have a global path, objects harmonically oscillate around an anchor. Professional tools often allow animators to create "brushes" of the elements to be animated, and manually adjust various parameters to specify turbulences. Such specialized controls are not supported in casual animation tools, in which case the animator is required to keyframe a set of example elements, then manually copy and paste them.

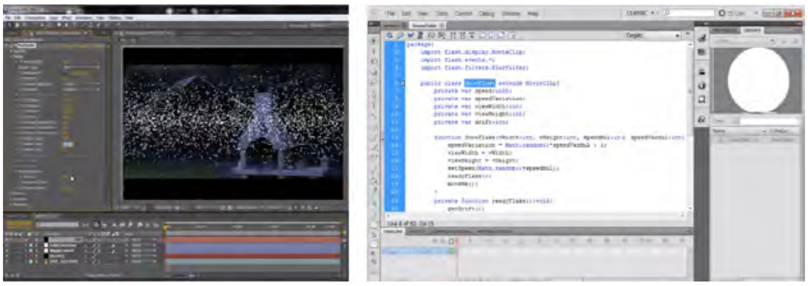


Figure 5-3: A dynamic illustration authored with Draco, capturing the living qualities of a moment with continuous dynamic phenomena, yet exhibiting the unique timeless nature of a still picture.

5.3.3 Observed Process

While the specific workflows can greatly vary, we did identify two high-level tasks common across animations of collections of objects.

Creating Collection of Objects

Collections of objects are typically generated by replicating a sample source object spatially. The typical workflow to replicate the source object involves manual copy-pasting, scripting (defining where and when objects should be replicated) or creating an emitter (in particles systems) to generate a continuous stream of objects.

Defining the Motion Trajectories

Objects in a collection are typically given motion at two levels of granularity: a global motion that applies to the entire collection and, within the collection, a granular motion that induces subtle variations in motion behavior across individual objects. Global motion is typically guided by a director path specified manually, or determined by the particles system emission direction. Local motion is achieved through keyframing, by using random variables in scripts, or by manipulating the particles system parameters.

5.3.4 Insights and Discussion

Our design study elicited several interesting insights that will be important to account for in our own designs.

One of the insights made from our observations is that animators have to use a number of different tools, techniques and workflows, depending on the type of animations they create. This limits the author's creative flexibility with any given tool.

It was also clear that while professional tools are extremely powerful and allow for production quality animations, specifying the desired behavior by the animator is still difficult. In particular, transposing a particular effect, even a very simple one, in terms of physics-based simulation requires significant expertise. This provides a significant barrier to novice users and clearly detracted from the overall experience even for experts:

P1: "I have to convert their artistic vision into physical parameters. I cannot provide input the way I am thinking."

Most importantly, the tedium associated with highly specialized physical simulations seemed to be a barrier to prototyping, brainstorming and creative exploration for animators. Our experts expressed their frustration for not being able to quickly try out effects as they come to mind:

P2: "I need the details of the whole shot before starting the animation, the trajectory, starting and ending points."

Taken together, these insights reflect the current need for rapid prototyping and exploration tools, to allow artists to quickly design, explore, and communicate animation effects involving collections of objects.

5.4 Design Goals

Based on the findings from the above study, we derived a set of design goals for our new system, which will support the rapid creation of scenes involving the animation of collections of objects.

Generality: Our system should enable users to create a variety of phenomena with a unified workflow. Unlike traditional tools, users should not need to be aware of specific simulation parameters. Furthermore, our system should not restrict users to specific pre-authored effects.

Multi-Scale Motion Dynamics: Our system should also support the authoring of motions at both the global and local scale of a collection. Global motion will control the overall shape and direction of the collection, while granular motion should direct the variations of individual elements.

Control & Flexibility: Our system should reduce tedium by synthesizing and propagating example motions to individual objects. Manually editing a collection of objects is too tedious due to the numerous elements and parameters, whereas fully automated motion computation has limited expressiveness. A mixed approach should offer generic control of motions, while supporting creative flexibility.

Simplified UI and Direct Manipulation: Our system should enable the creation and control of dynamic phenomena with relatively little effort by animators and amateurs, relying on users' intuitive sense of space and time with freeform sketching and direct manipulation.

Before describing our new system, which was developed to support these design goals, we first introduce the key components of our general animation framework.

5.5 Kinetic Textures: An Animation Framework

Based on the generalized workflow observed in our design study, we propose a framework built around kinetic textures, a novel animation component that encodes simple collections of objects and their associated motion. Our framework builds on general concepts that are easy to understand, while offering rich creative capabilities.

A kinetic texture consists of a patch—a small number of representative objects that serve as a source example to generate the collection, and a set of motion properties. The motion properties define the trajectory and movement of all the objects within the collection at two different scales: the global motion and the granular motion.

We introduce two types of kinetic textures: emitting textures and oscillating textures, which differ in how the collection is generated from the source patch, and how the global motion is defined (**Figure 5-4**). Emitting textures are motivated by particles systems and flocking, while oscillating textures allow the simulation of stochastic motion with repetitive, continuous harmonic motions.

5.5.1 Emitting Textures

Emitting textures are characterized by the continuous emission of a stream of objects (**Figure 5-4**a). Objects of the patch continuously emanate from the emitter, and follow a global motion trajectory, guided by the underlying motion vectors field computed from the motion path(s). Additional emitting textures components include the texture outline and mask(s), which can be specified to define the area of the animation. Objects decay as they cross the outline, and temporarily hide as they pass through a mask.

Three parameters control the dynamics of an emitting texture. The emission velocity controls the initial velocity of the objects, the emission frequency controls how frequently objects are emitted, and the cohesion controls the magnetism of the objects towards the motion paths (**Figure 5-5**).

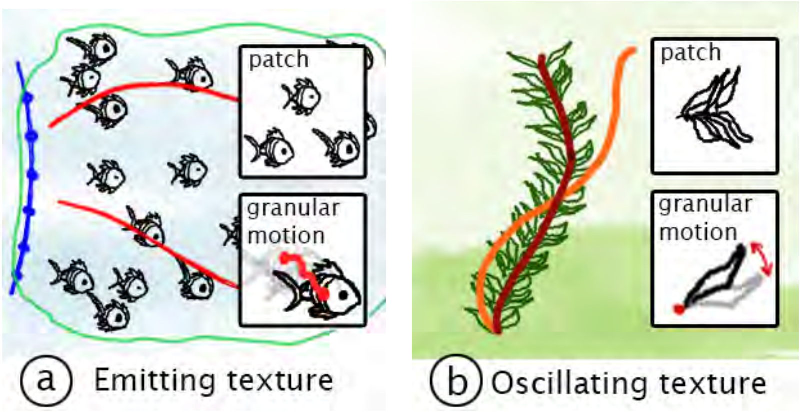


Figure 5-4: The two types of kinetic textures. (a) Emitting texture, defined by a source patch, emitter (blue), global motion paths (red) and granular motion. (b) Oscillating texture, defined by a source patch, brush skeleton (brown), oscillating skeleton (orange), and granular motion.

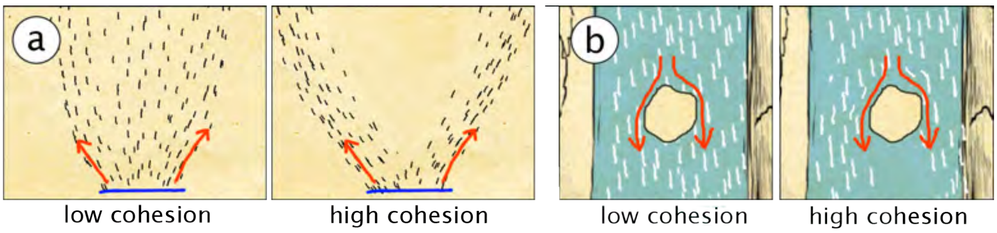


Figure 5-5: Impact of cohesion on the global motion. (a) A lower cohesion value produces a more uniform distribution between the motion paths. (b) Obstacle avoidance effects are obtained with a higher cohesion to the motion lines.

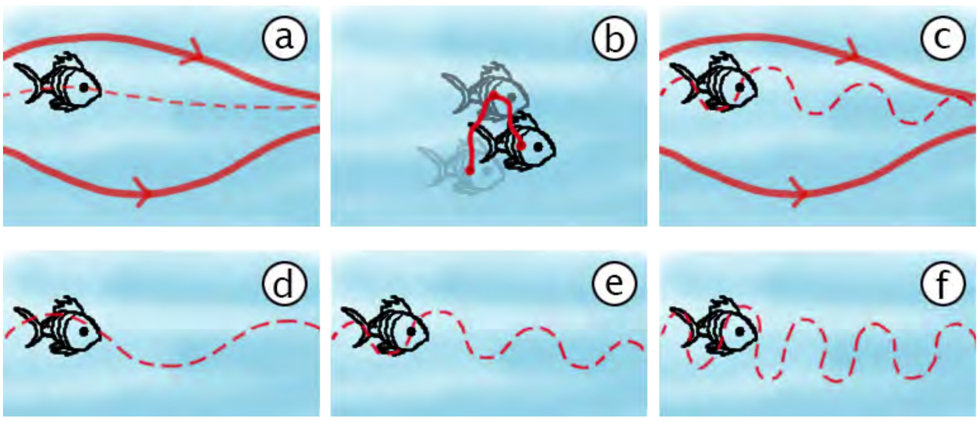


Figure 5-6: Motion factorization. Combining (a) the global motion trajectory and (b) the granular motion results in (c) the trajectory of individual objects. Manipulating the velocity of granular motion affects the object's trajectory (d-f).

5.5.2 Oscillating Textures

In contrast to emitting textures, an oscillating texture consists of a finite collection of objects, built by replicating a source patch along a brush skeleton. The global motion of the texture is characterized by the oscillatory movement of the objects along the skeleton between two positions (**Figure 5-4**b): the initial brush skeleton and a target oscillating skeleton. A parameter of the oscillating texture is the velocity at which the oscillations occur.

5.5.3 Granular Motion

The two types of textures described above define the global motions of the object collections. In addition, granular motions can be added for intricate and finer details. Granular motions apply to every individual object of the collection, and can either be a translation motion, where the objects move along a two-dimensional path (**Figure 5-4**a), or a pivot motion, where the objects rotate around a pivot point (**Figure 5-4**b). The trajectory and orientation of individual objects in the collection result from the combination of the global and granular motions (**Figure 5-6**).

Our framework provides two granular motion controls: velocity and phase. The velocity quantifies the frequency of the granular motion along the global path (**Figure 5-6**d-f). The phase refers to the level of synchronization of the granular motion among the individual objects. At minimum phase value, the granular motions of all objects are synchronized.

5.6 Draco: Design and Implementation

We designed and implemented Draco, a new system for the quick authoring of animations involving the coordinated motion of collections of objects (**Figure 5-7**), Draco builds on the above animation framework, and capitalizes the freeform nature of sketching and direct manipulation. The resulting animations are a juxtaposition of static strokes and kinetic textures. The interface contains a main authoring canvas, an interactive patch used to author granular motions, a tool palette (**Figure 5-8**), and a small set of basic parameter controls.

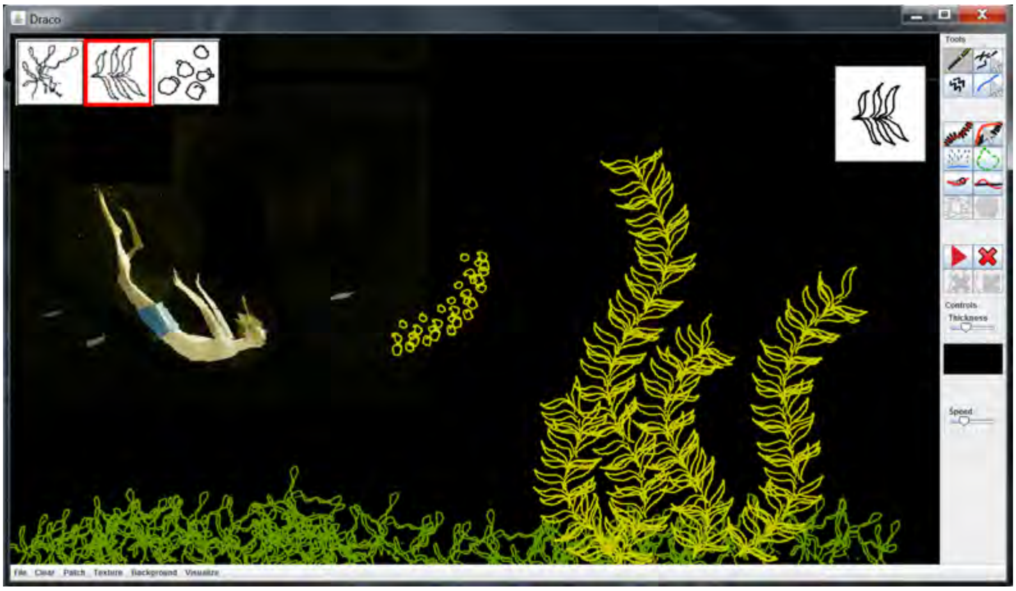


Figure 5-7: Draco user interface, consisting of a main canvas, an interactive patch, a tool palette, and parameter controls

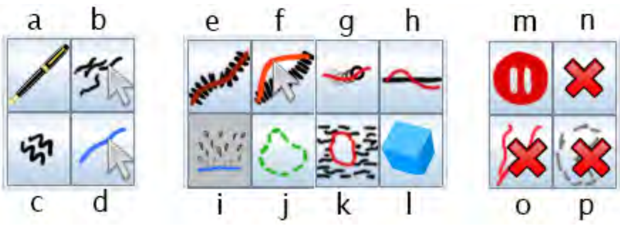


Figure 5-8: The Draco tools. (a) Ink, (b) Ink Selection, (c) Patch, (d) Patch Selection, (e) Skeleton Brush, (f) Oscillation, (g) Motion Path, (h) Motion Profile, (i) Emitter, (j) Texture outline, (k) Mask, (l) Perspective, (m) Play/Pause, (n) Remove texture, (o) Remove Motion, and (p) Remove Texture Mask

5.6.1 Interaction

Our design observations indicate two major repetitive tasks to add coordinated motion to collections of objects: creating the collection, and specifying their behavior (motion). We now describe the workflows for accomplishing these tasks.

Emitting Textures

Figure 5-9 depicts the different steps for creating an emitting texture. The user first selects the patch tool and draws a few representative objects that will compose the source patch to generate the target collection (**Figure 5-9**a). Using the emitter tool, she directly sketches the emitter by drawing a stroke on the main canvas (**Figure 5-9**b), after which the system immediately starts emitting elements perpendicular to the emitter (**Figure 5-9**c). If the emitter is a point, objects are emitted in all directions. The user can redraw the emitter by sketching a new emitter stroke, in which case the current emitter will instantaneously be replaced.

After defining the emitter, the user can adjust the global motion field of the collection by directly sketching motion paths on the canvas using the motion path tool (**Figure 5-9**d). The motion field is dynamically updated upon completion of each new motion path (**Figure 5-9**e). We provide further details on the computation of the motion field in the implementation section. Granular motions can subsequently be defined through direct manipulation with the interactive patch widget, described later (**Figure 5-9**f-h).

The user can also use the texture outline tool to sketch the boundaries of the texture, and the mask tool to sketch regions within which objects should be made invisible. Users can control the velocity, frequency, and cohesion of the emitting texture using associated sliders.

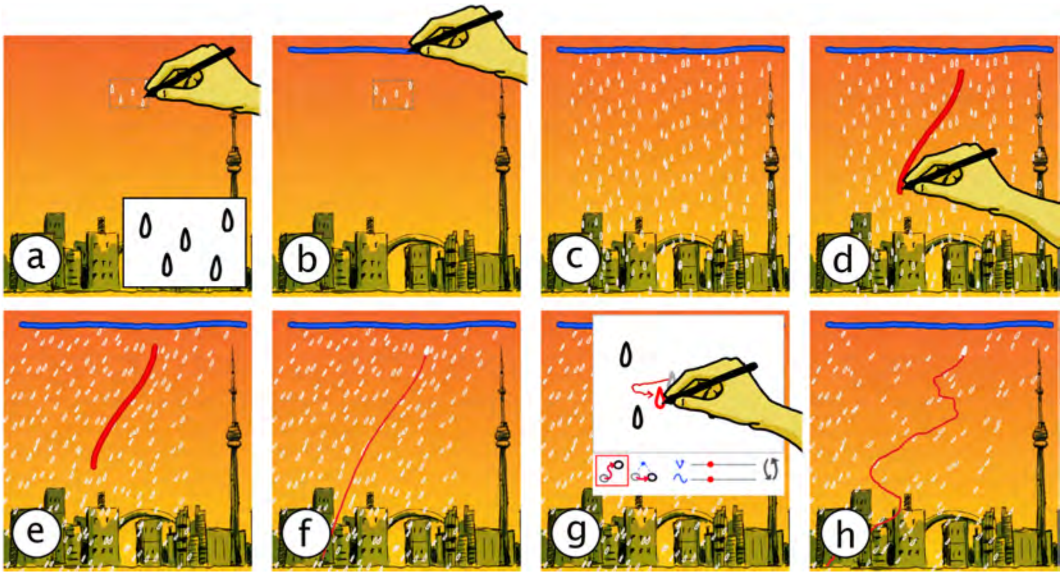


Figure 5-9: Creating an emitting texture. The user draws the source patch (inset: example raindrops) (a), then sketches a line emitter (b), which results in an emitting texture with a default motion (c). The user sketches a motion path (d), which instantaneously changes the global trajectory of the raindrops (e). Finally, she adjusts the granular motion by adding subtle translation to the raindrops (g), supplementing the global motion (f), with local variations (h).

Oscillating Textures

Figure 5-10 illustrates the workflow for creating an oscillating texture. First, the user sketches a few example objects using the patch tool (**Figure 5-10**a), then, with the skeleton brush, directly sketches the skeleton of the texture on the canvas (**Figure 5-10**b). This replicates the patch along the skeleton in a similar way as in the Vignette system [79] (**Figure 5-10**c).

To create an oscillatory motion, the user selects the oscillation tool, and sketches a target oscillating skeleton (**Figure 5-10**d). Upon completion, the texture oscillates between the two skeletons, interpolating the position and orientation of the repeated patch objects along the textured skeleton (**Figure 5-10**e). Similar as in the emitting texture, the oscillating skeleton can be redrawn by sketching the new form, which automatically updates the oscillation behavior. As with emitting textures, granular motions can subsequently be defined using the interactive patch widget (**Figure 5-10**f-g), described later. Users can control the speed of the oscillations using a slider.

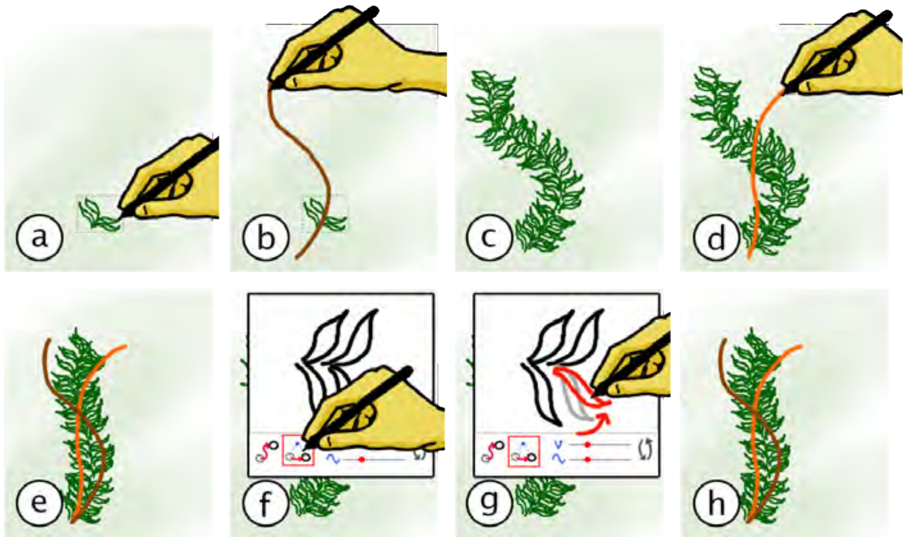


Figure 5-10: Creating an oscillating texture. The user draws the source patch (here example leaves) (a), then sketches the brush skeleton (b), which results in a brush texture, where the patch is replicating along the brush skeleton (c). The user sketches the oscillating skeleton (d), triggering the oscillation of the texture (e). Finally, she adds pivot granular motion (f-g), resulting in subtle local leave motions.

Granular Motion

As illustrated in the workflows in **Figure 5-9** and **Figure 5-10**, users can add granular motion to kinetic textures to induce local variation in motion to objects through the interactive patch widget. To add granular motion, the user first expands the patch region, then selects the type of motion: translation (**Figure 5-9**g) or pivot (**Figure 5-10**f-g). The user can then define the granular motion of objects through direct manipulation of any object within the patch. The performed transformation (displacement or rotation) is recorded as the user manipulates the example object, and is applied to all of the individual, repeated objects generated from the patch. The controls associated with granular motion are displayed below the expanded patch region, controlling the velocity and phase synchronization of the granular motion.

Motion Profile

When creating moving objects, it is often desirable to dynamically adjust their scale and velocity along their trajectory. For example, bubbles can grow and decelerate after their emission (**Figure 5-12**). To do so, the user selects the motion profile tool, which displays the profile widget at the bottom of the canvas (**Figure 5-11**). The user can then select either a scale or velocity icon and directly sketch the profile curve corresponding to the desired behavior. The height of the profile curve defines the scale or velocity of the elements along the associated point within its trajectory.

Figure 5-11: Motion profile widget. A motion profile curve can be sketched (a) to define the scale (b) or velocity (c) of the elements. The reference path along with marks is provided for guidance (d). Here, the scale is set to gradually increase as objects proceed along their path (see Figure 12f).

Workflow Flexibility

Our system was designed to be flexible in the workflows it supports. For example, when authoring an emitting texture, instead of defining an emitter first, the user could sketch a motion path, and a default emitter perpendicular to the motion path would automatically be defined. **Figure 5-12** depicts such an example. For an oscillating texture, granular motion can be added without defining an oscillating skeleton, to create a texture that has local motion only (e.g. skip steps d and e in **Figure 5-10**).

Furthermore, users can easily edit existing textures by overwriting components such as emitters, motion paths, motion profiles, and granular motions. This, combined with immediate visual feedback of the result of the user's actions greatly facilitates exploration, since the user can quickly experiment with different effects, with relatively little effort. When a texture is selected, its components are made visible with color-coded strokes (e.g. emitter drawn in blue, motion paths in red, brush skeletons in brown), providing the appropriate visual feedback to the user.

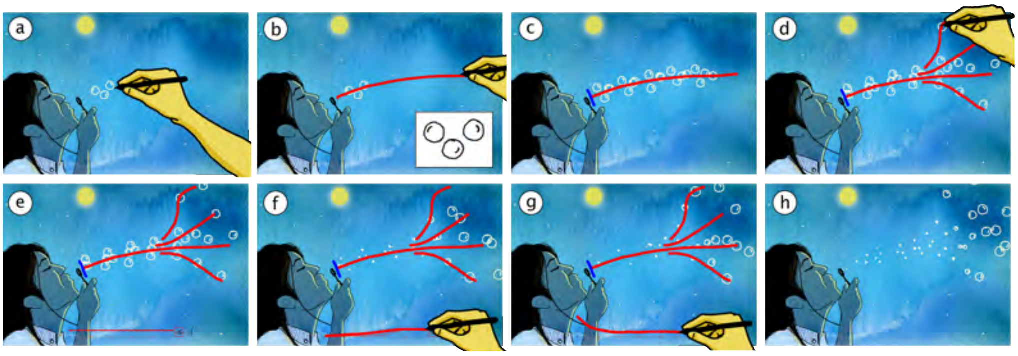


Figure 5-12: Creating an emitting texture. The user draws the sample objects (a), then directly sketches the motion path (b). An emitting texture is automatically created with a default (blue) emitter, perpendicular to the motion path (c). The user then sketches additional motion paths in order to spread out the bubbles (d). Finally, she uses the motion profile widget (e) to adjust the scale (f) and the velocity profiles (g), so that the bubbles grow and decelerate as they move away from the emitter.

Additional Features

Draco provides a number of features for interactive refinement and finer details.

Static Ink. Our resulting illustrations consist of both kinetic textures and static ink strokes. The Ink Tool enables users to sketch static strokes, which can be selected by lassoing with the Ink Selection Tool, and deleted with the Delete button.

Background Images. In addition to sketching static ink, users can import a static background image to sketch on top of. User can select from a set of pre-authored backgrounds, or choose any image from their own file system.

Visual Attributes. A color widget and a stroke size slider allow users to manipulate the visual attributes of both ink strokes and kinetic textures.

Perspective Tilting. Users can tilt a kinetic texture to create a 3D perspective effect with the Perspective tilt tool.

Texture Selection. By default, the texture currently being authored can be edited.

At any time, users can access, edit, or remove previously authored textures. Clicking on the canvas with the texture selection tool selects the texture associated with the closest emitter, or brush skeleton.

5.6.2 Implementation Details

Draco was implemented as a Java application. Our tool is multiplatform, and can run on any tablet or tablet pc.

Emitting Texture

We compute the global motion field from the motion paths following a similar algorithm developed by Chen et al. [21]. Each motion path is assigned discrete points P_m at fixed intervals, with their associated unit motion vector V_{Pm}. V_P denotes the direction of the global motion of an object at point P, which is defined as the weighted sum of all the motion vectors V_{Pm} as follows:

$$V_P = \sum_{P_m} \frac{1}{d^{\alpha}_{P,P_m}} V_{P_m}$$

where d_P, P_m is the distance between the current object location P and the motion path points P_m. The coefficient α defines the cohesion (magnetism) of the motion paths in the motion field (see **Figure 5-5**). The greater the value, the more the objects tend to be attracted by the motion paths.

Oscillating Texture

We use a simple harmonic oscillation to simulate the global motion of oscillating textures, using a sinusoidal curve in between the two skeletons. We use Fernquist et al.'s stroke guidance algorithm for morphing the shape of the curve between the brush skeleton and the target oscillating skeleton [44].

5.7 User Evaluation

We conducted a user evaluation with both professional animators and amateur illustrators, to gain insights about our animation framework, interaction techniques, unique capabilities, limitations and potential applications of our tool. This study is also used to gather insights on how our system compares to existing approaches, although we do not perform any sort of formal comparison to existing commercial tools.

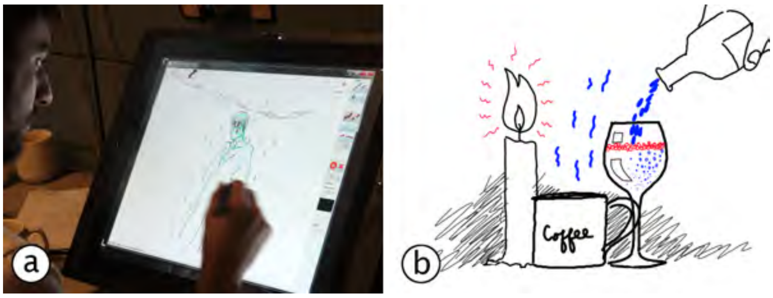


Figure 5-13: (a) A participant using our system to create animated illustrations. (b) The exercise task consisted of three emitting textures (blue) and two oscillating textures (red)

5.7.1 Participants

Eight participants took part of the study (7 males), aged 24 to 43 years old (average 32), half of which had moderate to good sketching and illustration skills (P1-P4) and four professional animators (P5-P8). Each participant received a $25 gift card reward for his or her participation.

5.7.2 Study Protocol

All the experiments were conducted using the Wacom CINTIQ 21ux tablet display (**Figure 5-13**a). The evaluation period lasted for 60~80 minutes 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 and demonstration of the system. Then, the instructor walked participants through 6 training tasks that consisted of simple animated scenes, such as rain falling from a cloud, and seaweeds oscillating underwater. The training tasks were carefully designed to familiarize the participants with the user interface, features, capabilities and core concepts of our animation framework (i.e. emitting textures, oscillating textures, granular motion and motion profiles). While the facilitator guided the participants to follow the step-by-step instructions, participants were also encouraged to explore at their will, and ask as many questions as desired during this training phase.

Exercise Task (10~15 minutes). Participants were given an exercise scene, consisting of 5 kinetic textures to reproduce from a model (**Figure 5-13**b). The exercise task covered different types of effects, including 3 emitting textures and 2 oscillating textures. Granular motions and motion profiles were also required to complete the exercise task. Participants were prompted with the video of the target effects on a separate display, which they could refer to at any moment. The facilitator did not intervene unless the participant had trouble using the system. No time limit was imposed. The purpose of this task was to observe whether the participants could easily reproduce a target effect. The facilitator recorded the completion time of the task, and logged any errors that were made in the workflows.

Freeform Animating and Feedback (20~25 minutes): Finally, participants were free to explore the tool to create dynamic illustrations of their own. Once done with their artwork, participants were asked to fill out a questionnaire to provide feedback about the system.

5.7.3 Results and Discussion

Overall, the participants responded positively to the simplicity of Draco's interface and concepts. All appreciated the unique capabilities of our tool:

> *P5: "It is not an animated scene with events and interaction, but I love the way it builds up the moment with ambient motion, making illustrations more expressive".*

Participants particularly liked the ability to quickly create different kinds of effects, which can be tedious to achieve otherwise. Participant's average rating of the system's overall ease of usage was 4.63 out of 5 (min 4). Regarding the overall experience, P3 commented:

P3: "Simple animation process with enough tools to create detailed graphics and animations."

Feedback on Animation Framework

Participants found the core concepts of our animation framework to be both useful and easy to use (**Figure 5-14**). However, for the kinetic textures, the standard deviation is higher for "ease of usage". We believe that, the first time users require more practice and time to get used to of all the icons and different components of the animation framework. As indicated in the user study, users were often confused about the icons and components. We believe a better icon and visual design would make the system easier to use for first time users.

P3: "Overall… the concepts are straightforward. From my experience, there are only three basic concepts, which are texture, emitter and motion. The rest of the tools are for adding greater detail to the animation."

Participants also liked the multi-scale motion for finer details and variations:

P2: "[Granular motion] really lets me add visual complexity very easily, that would be incredibly time consuming to do otherwise".

Participants also liked the fluidity of the motion profile to quickly overwrite and manipulate the properties of objects:

P2: "The animation curve sketching is great … very efficient way to manipulate details."

Exercise Task Performance

We were encouraged to see that all the participants finished the exercise task without any assistance. On average, the exercise task was completed in 7:40 minutes (min 4:30 minutes, max 13:40 minutes). Across all 8 participants, 7 workflow errors were made. For example, one user created an emitting texture but then quickly realized that it should be an oscillating texture. In all cases, users were able to independently recover from the errors that were encountered. This was facilitated by our system's abilities to quickly redraw sketched content, such as motion paths, and to immediately update the animation effects.

Some participants were more meticulous than others, spending more time fine tuning the results. For instance, P7 took 13 minutes without encountering any errors, while P1 took 4:30 minutes with two errors. Overall, the outcomes of the exercise task confirmed the ease of usage of Draco, and the effectiveness of our training session.

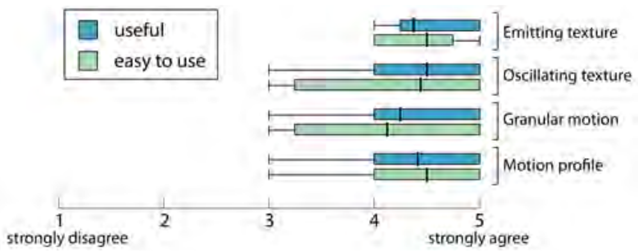


Figure 5-14: Subjective results for the four core concepts. The vertical bars indicate the highest and lowest ratings. The solid colored bars indicate the standard deviations.

Feature Usage and Artworks

Our participants authored a range of animated effects with kinetic textures in the freeform usage stage and post-experiment usage. Participants used oscillating textures to animate landscapes (e.g. trees, weeds), hair and simulating clothes and fires. Emitting textures were used for rainfalls, waterfalls, flocking, and water ripples. One participant used oscillating textures in an unexpected way, to animate the legs of a scuba diver (**Figure 5-15**a). Several participants used emitting textures to create a camera movement effect, with moving backgrounds (**Figure 5-15**b). We were pleased to see the system used in several ways that we had not previously considered, demonstrating the system's flexibility.

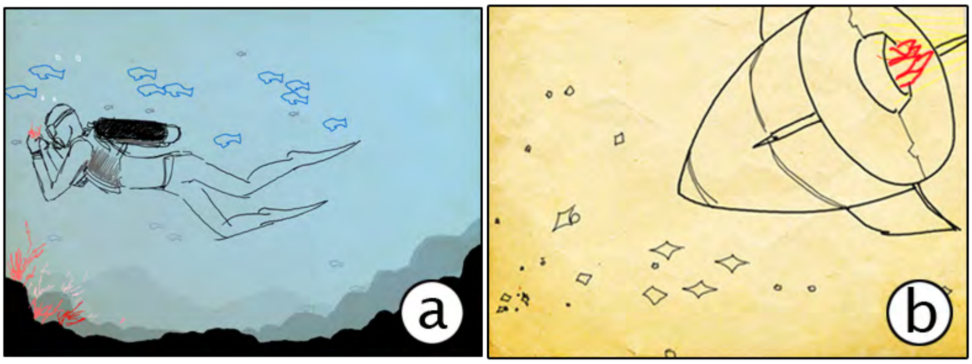


Figure 5-15: Artwork created by participants using Draco. (a) Underwater scene with leg movements (oscillating), school of fish and bubbles (emitting). (b) Flying rocket with flames (oscillating) and moving stars (point emitter / motion profile)

Potential Applications

Participants pointed out their desire to use our tool in a variety of applications, for both personal and professional usage. P2, P4 and P8 indicated motion and web comics as a suitable medium for kinetic textures:

> *P2: "I always wanted to have ambient motions in my comic panel backgrounds [...] unlike animations with events and actions that disrupt the experience of reading comics."*

P8 believes that our system could significantly reduce the tedium associated with creating animations for web comics. P4, P5, P7 and P8 pointed out to the fact that the system might be very appealing to children due to its ease of creation and playful experience.

> *P7: "I see this as a perfect tool for kids and family to use for sending out greeting cards with animated pictures. For professional usage, the first thing*

that came to my mind is animated illustrations in e-books for children."

All of the professional animators pointed out its potential application for brainstorming and communicating ideas during the storyboarding process.

> *P5: "It would be a great fit for Animatics (animated storyboards) to convey the results and ideas more clearly [...] Typically, storyboards consist of static sketches and simple animations (zoom) [...]. One can easily add particle effects to make it like a real scene."*

P3, a graphic designer, believes such kinetic textures can be used for authoring animated graphical objects to enhance web content such as dynamic cursors, icons, buttons and backgrounds, which is difficult to produce with Flash.

Two animators (P7, P8) mentioned that these kinds of tools might not fit with their current production pipeline due to tool dependencies, visual style and other constraints. However, both of them mentioned that such tools can be used in TV shows with certain visual styles and illustrating some ideas when pitching ideas to the clients. Other potential applications are photo collages (P5), animated diagrams for presentations, papers and videos (P1), as well as online portfolios (P3).

5.7.4 Limitations and Future Works

While participants were generally satisfied with Draco, they also pointed out some limitations that could guide future enhancements.

Most notably, participants wanted to instill ambient motion into single objects:

> *P2: "I have to think in terms of patterns (textures), rather than just animating a single object".*

We were aware that participant might see this as a limitation, but our research focus was on the animation framework for simple collections of objects. We believe systems like K-Sketch [34] could adequately address this limitation. In the future, it would be interesting to expand the vocabulary of our motion to be able to animate both structured textures and individual objects.

Another limitation pointed out by P8 was a lack of interaction between the objects (i.e. collisions, attraction) during the animation. In this project, we focused on ease of creation and real-time performance, rather than precision and physical accuracy. However, additional controls for object interactions would be a fruitful area of exploration.

Participants noted several other improvement opportunities and feature requests, such as: improved drawing capabilities with better rendering, brushes, and opacity (P3, P6, P8); the ability to provide more than two skeletons for oscillating textures (P6, P7, P8); finer controls for granular motions, such as deformations and changing pivot points for rotations (P1, P2, P5, P8); and changing object density spatially with motion profile like controls (P5, P7, P8).

In the future, we plan to give greater controls to more advanced users, without sacrificing the simplicity of usage. One way to achieve that goal might be to use a hierarchical user interface, where advanced users can initiate more advanced settings and controls according to their usage.

5.8 Summary

Draco is a sketching tool that enables the creation of a wide range of intricate animation effects, seemingly bringing illustrations to life. The core component of our system is kinetic textures, a new animation framework, which simultaneously achieves generality, control and ease of use. The interaction techniques within Draco capitalize on the freeform nature of sketching and direct manipulation to seamlessly author and control coordinated motions of collections of objects. Draco pushes the boundary of an emerging form of visual media that lies between static illustration and videos. Our user evaluation points to a variety of applications that would potentially empower end users to author and explore animation effects quickly and easily, and also suggests a number of interesting areas for future improvements.

Unlike SandCanvas and Vignette that are inspired by traditional art media Draco seeks design insights from exisiting computer animation tools and techniques. Draco extends the capabilities of Vignette by faciliting the creation of a wide range of animation effects with simple sketch-based interface.

However, we followed similar design process for Vignette. We started by looking at the artifacts, which is the scope of animations in this process. We interviewed experienced professionals and observed online tutorials to gain insights into the workflow and its pain-points. We identified key insights, which we formulated as design goals for the framework. Our user interface capitalizes the freeform nature of sketching and direct manipulation. Our user study suggests the potential of this medium in variety of application domains. Draco extends an existing digital art tool (Vignette) to a new temporal dimension, by seeking inspiration from decade long computer animation media. However, Draco stands unique from both texture illustration and computer animation media. The artifacts authored by Draco have continuous animation effects, which enhances our experience, yet, it preserves the unique timeless nature of static illustrations.

6. SKETCHSTORY: AN ENGAGING TOOL FOR STORYTELLING WITH DATA THROUGH FREEFORM SKETCHING

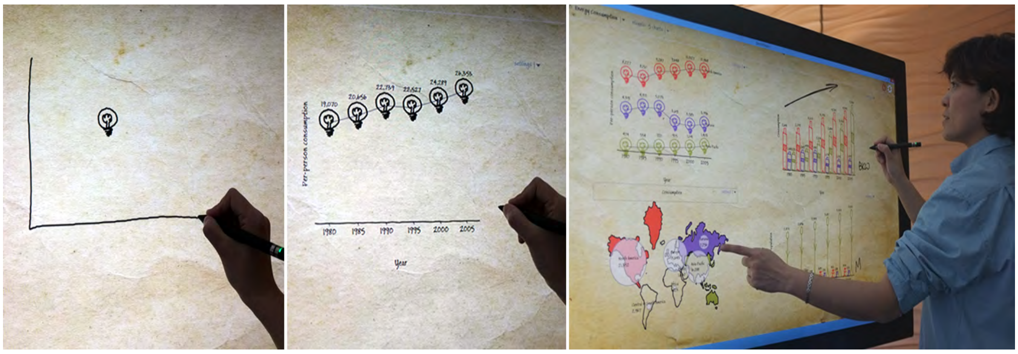


Figure 6-1: Telling a story using SketchStory: (a) The presenter sketches out example icon and chart axis, (b) Upon recognition of the chart axis, SketchStory completes the chart with underlying data by synthesizing from example sketches, and (c) Presenter interacting with the charts.

In the previous chapters, SandCanvas is a medium particularly tailored for real-time storytelling and performance art. While, Vignette facilitate the creation of personalized texture drawings by synthesizing from example strokes. SketchStory enables users to perform real time presentation with data by synthesizing example sketches in the whiteboard to create charts in real time. SandCanvas was inspired by sand animation, Vignette was inspired by pen-and-ink illustration and Draco extended the capalities of Vignette into temporal domain inspired by digital animation tools and techniques. However, multiple popular traditional art communication media – whiteboard animation, infographics and interactive information visualization techniques, inspires the design of SketchStory.

6.1 Background and Motivation

One of the main goals of Information Visualization (InfoVis) is to help people gain insights—finding underlying patterns and relationships between bits of data hidden by raw quantity—more easily through their innate perceptual and cognitive capabilities. Accordingly, over the last two decades, the InfoVis research community has focused on developing techniques and systems that facilitate gaining insights by representing abstract information in interactive, visual forms. Although effectively presenting or communicating these insights to others is often the ultimate end-goal of data exploration, until recently insight presentation and communication has been relatively less explored by the InfoVis research community. But there is now a growing interest in novel forms of storytelling techniques with data, commonly known as narrative visualization [77]. As an emerging medium, narrative visualization can borrow techniques from existing storytelling sources (e.g., comics, posters, etc.) [77], and extend them to develop a more engaging form of storytelling.

In his well-known book, "The Back of the Napkin," Dan Rom states that people like seeing other people's pictures, and that, in most presentation situations, audiences respond better to hand-drawn images (however crudely drawn) than to polished graphics [116]. The popularity of whiteboard animation (also known as video scribing) [117] is good evidence for his claim. In whiteboard animation, the presenter produces a sequence of dynamic sketches along with synchronized narration to vividly tell a story. The narrated, animated content creation and expressive graphic style makes whiteboard animation a very unique and engaging form of storytelling. As such, it has increasingly attracted both audiences and artistically-inclined presenters, and has become increasingly popular in domains such as advertising and education (e.g., [31][117]). However, producing high-quality whiteboard animation is time-consuming and potentially expensive; furthermore, its power to communicate with data is limited by relying purely on the presenter's numeric and artistic ability to formulate and depict the underlying numbers in a visually compelling way during a live performance.

This project makes the following contributions.

- To create a novel and more engaging storytelling tool with data, this project explores how to leverage and extend the narrative storytelling attributes of whiteboard animation using pen and touch interactions. This chapter presents SketchStory (**Figure 6-1**), a data-enabled digital whiteboard specifically designed to support telling more engaging stories with data through freeform sketching. It facilitates the creation of charts in real time by synthesizing from the presenter's sample sketches, preserving the expressiveness and organic style of visual graphics. SketchStory helps the presenter stay focused on telling her story by eliminating the burden of manual data binding. It allows the presenter to record a sequence of charts along with example icons before the presentation and to invoke them with simple sketch gestures in real-time. Furthermore, it enables the presenter to add freeform annotation and to interact with the charts created during the presentation. This helps invite discussion, explanation, and further exploration.

- Pursuing an iterative design strategy, we first conducted a formative study with six presenter participants. This helped us understand the usability and unique affordances of the SketchStory approach for presenting a story with data. We improved the system based on the lessons learned from the usability study. We then conducted a controlled experiment to compare SketchStory with Microsoft PowerPoint [114], one of the most commonly used presentation tools, for both the audience and presenters. Results show that the audience is more engaged with the presentation done with SketchStory than PowerPoint. Eighteen out of 24 audience participants preferred SketchStory to PowerPoint. Four out of five presenter participants also favored SketchStory even while acknowledging the extra presentation effort it required. In addition to these promising results, we identified future research directions gleaned from both studies.

6.2 Storytelling with Information Visualization

Storytelling allows visualization to reveal information effectively [132]. On the other hand, as Wojtkowski and Wojtkowski pointed out, it can be very effective to tell stories with data visualization [70]. Therefore, storytelling with data has begun to gain more attention as storytellers integrate visualizations into their narratives. The InfoVis research community has organized a workshop "Telling Stories with Data" two years in a row (2010 and 2011), focusing on exemplars of stories told with data and the techniques used to construct the stories. Recently, Segel and Heer reviewed the design space of this emerging class of visualizations called narrative visualizations [77]. They identified seven genres of narrative visualization—magazine style, annotated chart, partitioned poster, flow chart, comic strip, slide show, and video—from an analysis of 58 examples. Within this characterization, SketchStory spans the genres of annotated chart, partitioned poster, and video.

Several visualization systems have been incorporating storytelling into their design, primarily through graphical history and annotation. For example, the sense.us system provides bookmark trails, a graphical list of bookmarks, with graphical annotation to support storytelling [65]. Tableau's graphical histories allow people to review, collate, and export main insights of their visual analysis [105]. Recently, Tableau Public supports the publication of interactive visualizations on the web, enabling storytelling with data visualization [151]. GeoTime Stories enables analysts to create and present annotated stories within visualizations using a customized text editor for a story document containing links to visualization snapshots. These systems augment their exploration function by providing storytelling capabilities through an extension, mainly for asynchronous storytelling. However, SketchStory was specifically designed for more engaging, real-time storytelling as a main goal.

6.3 Medium, Artifacts and Process: Seeking Design Insights

The primal goal of this project was to explore a novel genre of narrative visualization technique specifically for storytelling and presentation. Hence we assume that the presenter has already found a story to tell during a prior exploration phase. We looked into the strengths and limits of popular and pervasive storytelling techniques with data, and explored ways to harness the strengths and to overcome the limitations.

While the previous work seeks design inspiration from an existing medium, this work seeks inspiration and insights from three existing popular medium- *infographics* for expressiveness, *whiteboard animation* for real-time content creation and *interactive information visualizations* for interactivity.

6.3.1 Medium: A New form of Storytelling with Data

Data visualization melds the skills of computer science, statistics, artistic design, and storytelling to make massive amounts of data more easily accessible. However, it remains an open question how to support richer and more diverse forms of storytelling with data [77]. In this section, we identify three desirable properties of storytelling with data; expressiveness, narrative sketching, and interactivity.

Expressiveness in InfoGraphics

Information graphics (or infographics) are graphical representations of information, data, or knowledge. Infographics are commonly used by reporters, computer scientists, and statisticians for communicating conceptual information in a broad range of domains. To clearly communicate complex information in an aesthetically pleasing way, they often employ icons and other visual elements that are customized to the dataset (**Figure 6-2**).

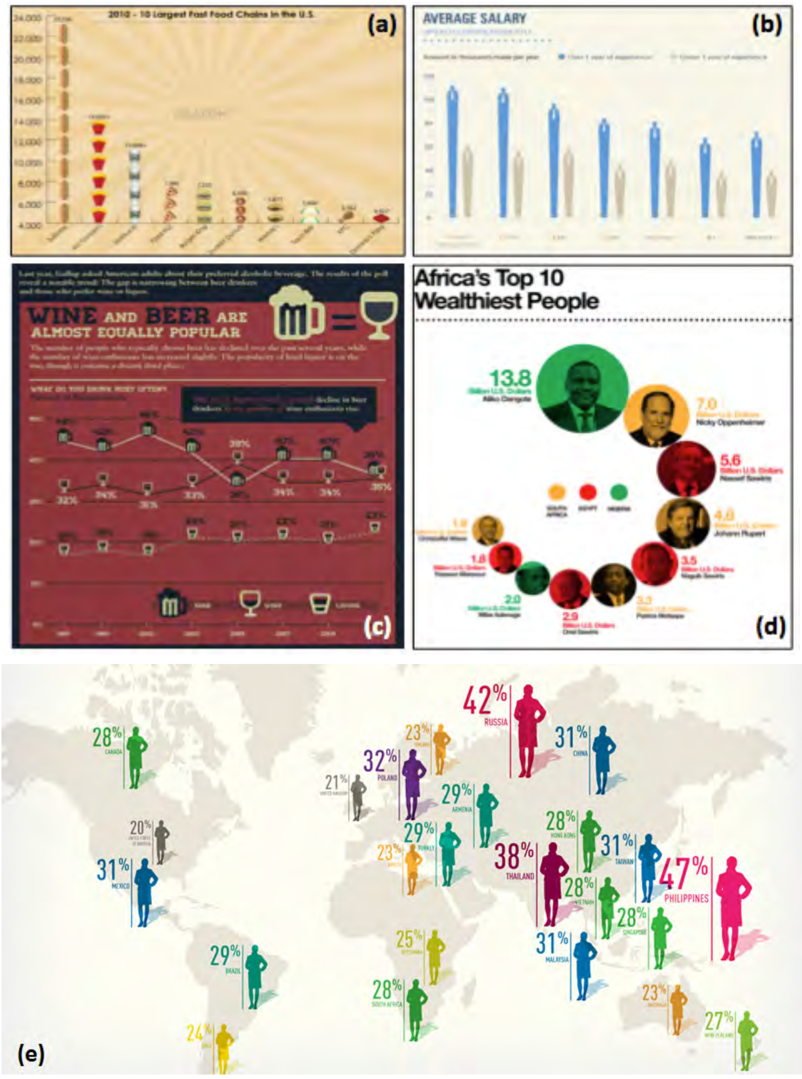


Figure 6-2: Examples of expressive and iconic data representation in infographics: (a) A tally chart shows the consumption of fast foods using food icons that represents each food type, (b) A bar chart uses a corporate executive icon instead of the traditional rectangle to represent average salaries, (c) A line chart with beverage icons conveys the fact that wine and beer are almost equally popular, (d) Custom picture icons are used to display an individual wealth metric, using a non-traditional chart layout with a circular baseline, and (e) Custom icons representing geographic data overlaid on a map

We believe the communicative power of customized infographics stems from two key factors. First, a close mapping between the graphical representations and the underlying data helps people make connections between them and facilitates understanding. Prior neurological studies indicate that fictional, metaphoric representation of facts and narratives activate many other parts of the brain [29]. Despite some criticisms [119] of Bateman et al.'s methodology, their study suggests that people's recall of embellished charts (after a two- to three-week gap) is significantly better than that of plain charts without sacrificing description accuracy [130]. Second, a custom visual design allows presenters the artistic freedom to create a unique, personalized chart taking full advantage of an innate visual language that is largely universal [116].

Often created with sophisticated graphical tools such as Photoshop and Illustrator, these visualizations can be both aesthetically pleasing and highly expressive. On the other hand, they are largely static, missing out on the full breadth of communicative power available to a live storyteller.

Narrative Sketching in Whiteboard Animation

Whiteboard animation is another compelling visual communication technique, where the presenter simultaneously narrates and sketches a sequence of line art elements to vividly tell a story (**Figure 6-3**). Like infographics, whiteboard animation builds on visual explanation with expressive graphics, yet it augments the storytelling aspect by linearly and verbally developing the graphical elements. Viewers consume the content one step at a time, following a logical sequence that makes the story easier to understand. Whiteboard animation is becoming a more memorable and effective method for delivering information, increasingly attracting audiences and storytellers and getting more popular for commercials, communications, presentations, and tutorials (e.g., [31][117]).

Several attributes of whiteboard animation make it a memorable, effective, and engaging method for visual communication. First, the dynamic re-enactment of a presenter’s sketch (or whiteboard presentation) conveys the order of action sequences by directing viewer attention from one object to the next, building anticipation as in a story. Second, skilled hand-drawn sketches generate organic and expressive graphics, allowing a personal, unique storytelling process. Viewers of sketched stories are inclined to focus on the important aspects such as the overall structure and flow since they tend not to focus on unnecessary details (e.g., precise font size, alignments) [7][104]. Third, as with other performance art media, the process of creation drives attraction and aesthetic appreciation. The storytelling attributes of whiteboard animation go beyond entertainment to engagement, making it an effective medium to plant ideas, emotions, and thoughts in viewers’ minds [76]. Fourth, the use of real-time narration and a canvas displaying the full visual interaction history provides constant contextual information to augment the communicative process.

However, creating traditional whiteboard animations is expensive and labour intensive; individual lines and text all need to be manually drawn, requiring intensive editing and post-processing of the recorded video. Any quantitative data presented visually is, in effect, “made up” by the artist during the drawing process rather than being backed by the underlying numbers in a more formal way. Finally, even though whiteboard animation is perceived to be more engaging than the regular video, it is still not interactive.

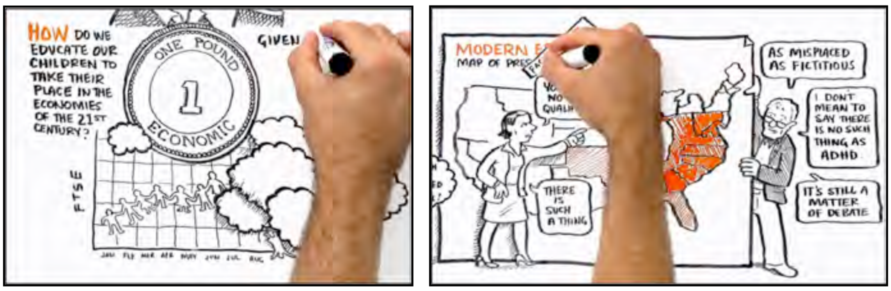


Figure 6-3: Manual sketching of data representation in whiteboard animation

Interactivity in Information Visualization

Interaction plays a critical role in information visualization particularly for exploring large and complex datasets. For example, dynamic queries [43] are one of the most commonly used interaction techniques for interactive visualizations. They enable people to formulate queries by manipulating embedded widgets (e.g., check boxes), and immediately see the query results. Also, information visualizations often combine multiple views through interactive linking and brushing to enhance the individual visualizations [68]. Since changes made in one visualization are automatically reflected in other linked visualizations, more information can be gleaned than considering the component visualizations independently [79].

Furthermore, interactivity in storytelling with data invites verification, further questioning, and exploration of alternative explanations [77]. Recognizing the importance of this interaction, there are on-going efforts to make infographics interactive. For example, news organizations such as the New York Times and Washington Post employ dynamic infographics on their websites.

Also, tools like Many Eyes [152] and Tableau Public [151] enable publishing interactive visualizations on the web more easily. But these efforts do little to aid narrative communication of the story. While general purpose, low-level rendering APIs (e.g., Java2D, Processing) are also available, construction of even simple charts is tedious and they lack narrative communication as well.

6.3.2 Design Rationale

Our goal was to create a new, more engaging way of telling stories with data by inheriting and extending the advantages of the successful storytelling techniques described in the previous section. In this section, we describe our design rationales behind the design of SketchStory.

Support real-time content creation with animated sketch

With traditional presentation tools such as Microsoft PowerPoint and Apple Keynote, the audience is often visually exposed to content significantly before the presenter verbally presents it unless she uses sub-slide appearance animations. To attract attention and create anticipation, SketchStory uses a real-time approach to content creation like many other performance art techniques do [20]. In addition, SketchStory combines the expressive visual language of infographics and dynamic sketch with narrative communication of whiteboard animation. To enable the presenter to generate expressive and custom representations that can be better aligned with her narration (i.e., story), SketchStory lets her provide any desired representation as an example visual element. It then synthesizes a full chart from the example sketch. Furthermore, SketchStory completes the chart with fluent animation to make it look like the presenter is sketching at a faster speed.

Aid Narrative Communication by reducing manual burden

It is burdensome to manually draw an entire presentation or to interactively specify each chart setting (e.g., chart type, axis, etc.) during the presentation. To alleviate this burden, SketchStory's design enables the presenter to pre-specify a sequence of charts and invoke each one with sketch interaction in real-time. Furthermore, to reduce the burden of manually drawing each icon during the presentation, SketchStory allows the presenter to optionally pre-record an example icon for each chart and save it with the other chart settings. The presenter can focus on narration while SketchStory takes care of the visual presentation at whatever detail level the presenter wishes.

SketchStory recognizes a small number of sketch gestures for creating (or invoking) different types of pre-specified charts. For example, the presenter can draw an 'L' shape to invoke a chart with x and y axis, or draw a circle to invoke a pie chart (**Figure 6-4**). SketchStory also recognizes touch gestures for moving and sizing charts. For example, the presenter can resize a chart with a one- or two-hand pinch gesture. Previous research shows that, when both pen and touch interactions are supported, people clearly distinguish between appropriate pen and touch interactions [64][94]. SketchStory leverages this by using the pen for drawing charts or annotations and touch for manipulating them, thereby avoiding having two explicit modes for sketching and manipulation.

Provide Interactivity and Contextual Information

In contrast to most traditional presentation tools, SketchStory uses the notion of a canvas to present information, and supports freeform annotation anywhere on the canvas for emphasis and decoration. This is to help the audience understand the context of the whole story and derive relationships between the visual components. Furthermore, to make storytelling more dynamic and responsive, SketchStory supports interactivity—missing from infographics and whiteboard animation—by tightly coupling the data displayed in all the charts on a canvas and allowing data filter changes in real time.

6.3.3 Choice of Data Charts in SketchStory

To inform the design of data charts in SketchStory, we examined the common data charts used in infographics. There are disparate sources for infographics examples including data art websites, visualization blogs, newspapers and scientific articles. To avoid subjective selection, we extracted the first 100 search results for the keyword "infographics," from two search engines (Bing and Google, dated 24th December, 2012). Of the resulting 187 infographics (13 appeared in both results lists), 149 (79.6%) of them depicted numeric data. Each infographic consisted of one or multiple graphical elements representing data: charts, symbols, stylized text, or customized visuals. We tabulated each of these elements; a total of 795 graphical data representations were logged in these 149 infographics (on average, 5.3 data representations per infographic).

We grouped the elements into 9 categories—bar (20.8%), pie (18%), tally (9.4%), scale (8.8%, e.g., Figure 2d), stacked bar (6.8%), map (5%), line (2.9%), area (2%), and tag cloud (1.1%). The remaining 25% were too customized to fit into any of these categories; they were tabulated as custom. For all data charts, we also logged them as custom iconic vs. standard representation. Despite the fact that iconic data charts require graphical tools (Photoshop, Illustrator, etc.) and can be laborious to create, we found 24% of the data charts to be iconic, which further motivates our design goal to facilitate the creation of iconic data charts.

We began by incorporating the three most popular chart types (bar, pie, and tally) into SketchStory, and included line chart and scatterplot because they were straightforward variants of the bar chart. We also included map, both because we wanted to explore the particular expressiveness of maps and because we wanted to be able to use maps as interactive objects for dynamic filtering.

6.4 SketchStory

As mentioned above, SketchStory is specifically designed for storytelling and presentation. Once the presenter has found a data-based story to tell via data exploration, this story can be prepared in SketchStory as a sequence of charts, where each chart specifies settings such as chart type, data columns, and potentially a pre-recorded example icon. During presentation, this sequence is available to the presenter as a dropdown of chart thumbnails (**Figure 6-4**, right), where the next chart in the sequence is indicated by a visible check mark.

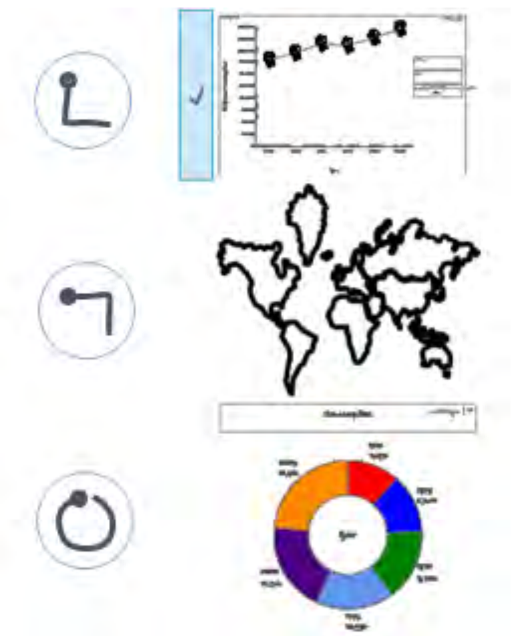


Figure 6-4: Preview of the part of the sequence of charts for the energy consumption story (right) and sketch gestures to invoke them (left)

6.4.1 Interaction

Creation of Expressive data charts

The interaction for chart generation consists of two simple steps; the presenter 1) sketches an example icon (**Figure 6-5**a) and then draws the sketch gesture (**Figure 6-5**b, **Figure 6-4**, left) in the desired size and location for the desired chart. SketchStory recognizes the sketch gesture and automatically completes the chart according to the chart settings specified in the presentation sequence (**Figure 6-5**c). To support the case where the presenter does not want to use an iconic representation, the first step is optional. If the presenter does not provide an example icon either during the presentation or in the chart settings, SketchStory creates a standard chart (**Figure 6-5**d).

The four chart types that involve x and y axes (bar chart, tally chart, line chart, and scatterplot) are invoked using an "L" sketch gesture; the pie chart is invoked with a circle gesture; and the map is invoked with a rotated "L" gesture (**Figure 6-4**).

Interactivity through visual keywords

SketchStory supports dynamic filtering, a very common interaction technique in InfoVis, through visual keywords (i.e., shaped stroke or icon). The presenter can create a mapping between a set of strokes and a textual keyword before the presentation, in order to perform dynamic filtering during the presentation by interacting with the strokes instead of entering the keyword. Visual keywords can depict an existing geographical map, or an iconic representation of keywords, to preserve the expressive graphical style of the visuals. For example, **Figure 6-6**a uses seven visual keywords to represent seven different regions (e.g., North America, Europe, etc.) and **Figure 6-6**b creates a mapping between two visual keywords that represent two genders—female and male.

When the presenter selects a visual keyword by tapping the icon (**Figure 6-6**c), SketchStory updates other charts on the canvas according to the keyword associated with that selected icon (**Figure 6-6**d). SketchStory toggles the selection when the presenter taps the visual keywords.

Freeform annotation and chart management

SketchStory supports freeform annotation anywhere on the canvas because it is useful for explanation, emphasis, and decoration. For example, the presenter can write down the unit for data values or draw an arrow to emphasize the trend (**Figure 6-6**d). SketchStory also allows the presenter to move and resize charts with touch interaction. For example, the presenter can make a chart bigger with pinch gestures to focus on the chart. When the presenter moves a chart, the annotations drawn within the chart boundary move with the chart.

6.4.2 Implementation Details

SketchStory was implemented in C# using WPF (Windows Presentation Foundation) and handles both pen and touch input. It runs on any pen and touch enabled Windows device since it relies on standard input event handlers.

On initialization, SketchStory loads all data files in a designated folder (which can be changed) and provides the available data stories in a dropdown control. SketchStory consumes a tab-delimited format for its raw data files, where each row of input is one data item and the tab-separated values represent the item's column values. Each data file requires an additional companion XML metadata file that describes the ordering, name, type (e.g., ordinal or numeric), and optional map files for each column. This metadata file also contains an optional chart sequence, where individual settings are serialized, specifying chart type, chart data columns, chart icon strokes, and other visual chart options for each chart in the sequence.

A map file is an additional XML document consisting of a series of pen strokes with associated visual keywords that correspond to column values for a particular data column. Strokes are persisted in the form of XML polyline objects as a sequence of x, y points.

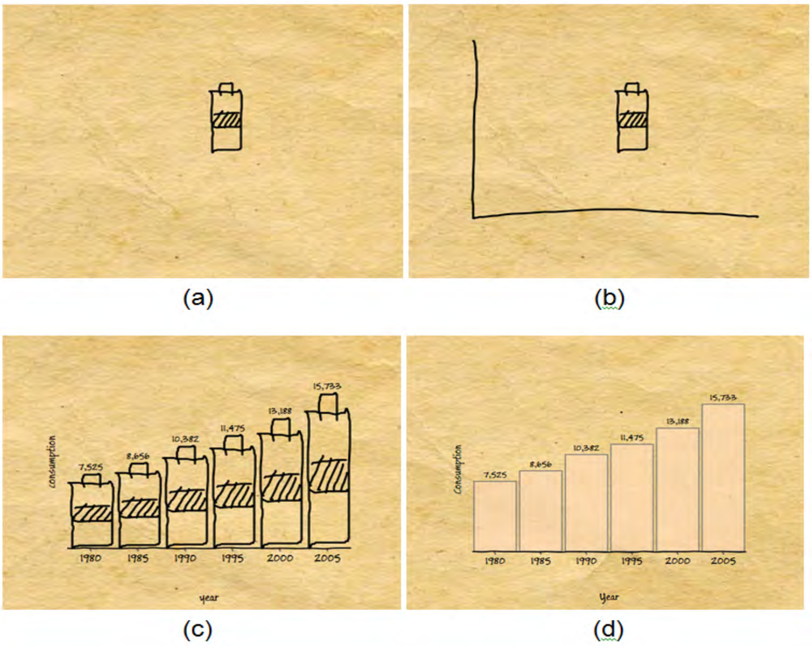


Figure 6-5: Creation of a chart: (a) Drawing an example icon, (b) Drawing an axis, and Automatic completion of the chart by SketchStory with an example icon (c) and without an example icon (d).

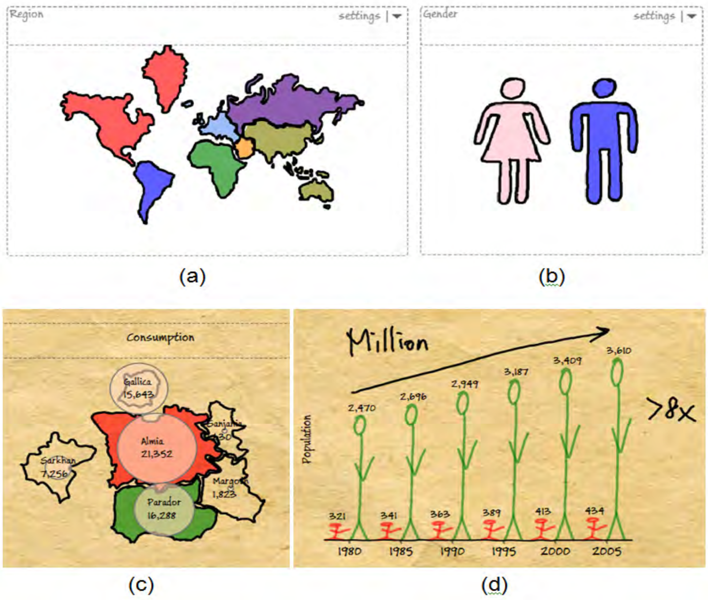


Figure 6-6: Visual keywords—mapping between strokes and keywords—for dynamic filtering, and annotation: (a) region map to represent seven different regions, (b) gender map to represent female and male, (c) scale chart overlaid on a fake region map used for the user study, and (d) chart with annotation.

Once the data is loaded, whenever the user adds a stroke to an empty area of the canvas with the pen, SketchStory attempts to segment the stroke into a polyline using a corner-finding algorithm [131] and then matches the stroke against the chart-invocation gesture set. If there is a match, SketchStory fills in the chart's data according to the next settings object in the current data story's chart sequence and renders the completed chart with animation. If the chart invocation gesture surrounds an existing set of strokes on the canvas, that set is cloned and used as the example chart icon.

6.5 User Evaluation

We conducted two studies to evaluate SketchStory. We first ran a preliminary usability study to identify major usability issues and to investigate unique affordances of SketchStory for presenters. We improved the SketchStory interaction based on the lessons learned from the first study. We then conducted a controlled experiment to compare SketchStory with PowerPoint, a traditional presentation system, in terms of subjective level of engagement for both the audience and presenters.

6.5.1 Usability Study

Initial SketchStory prototype

SketchStory initially focused on stories about trends (i.e., changes over time), relying on time (e.g., year) as the default x-axis. The SketchStory interaction for chart generation consisted of three modal steps. First, the presenter selected the desired value attribute (i.e., the numerical data for the y-axis) from the data attribute menu; the currently selected data attribute was shown under the menu. Second, the presenter switched to an example icon mode and sketched one or more example icons. Finally, the presenter switched to a chart axis mode and drew the baseline for the desired chart.

SketchStory initially supported four types of data charts; tally chart, bar chart, line chart, and map. After the presenter sketched the baseline in the third step, SketchStory clustered the example strokes into icons. Based on the example icon and baseline stroke, SketchStory automatically generated a chart using heuristics (e.g., a line chart if a single icon does not intersect with the baseline or a bar chart if the single icon intersects with the baseline).

Participants and Study Setup

We recruited six participants (3 females) comfortable with sketching from an industrial company. P1, P2, and P5 were UX (user experience) designers; P3 and P4 were program managers; and P6 was a researcher. Participants were also proficient with digital design and presentation tools such as Photoshop, and PowerPoint. They received $20 worth of lunch coupons for their participation. We conducted the study on a 3.2 GHz Windows 7 desktop machine with 12 GB RAM and a 27" Perceptive Pixel Display [181] that supports both pen and touch interaction at a resolution of 2560x1440.

Procedure

At the start of each session, we asked participants to fill out a pre-experiment questionnaire to collect their background about sketching, design, and presentation skills.

Then we gave the participants a brief introduction to SketchStory with a printed tutorial. The 10–15 minute training was broken into two phases. In the first phase, the participants sketched different types of charts and visual elements, following the step-by-step illustration of the tutorial. This phase familiarized the participants with the interface, features, and different types of visual elements and data charts within the scope of SketchStory. In the second phase, the participants replicated an example presentation to convey a simple story to the experimenter with a training dataset. This phase familiarized themselves with the storytelling aspects of SketchStory. The experimenter did not intervene unless a participant had trouble using the system.

After the training, we directed participants to a new dataset–global energy consumption data between 1980 and 2005 [31]. We asked the participants to tell a story with this data around the following key points:

- Global energy consumption doubled.
- Global population increased but less than 50%.
- Per-person energy consumption has also increased.
- North America and Asia-Pacific are the top two consumers.
- While Asian population is more than 8 times higher than North America's, on average American person consumes more than 7 times more energy than Asian person.

In order to familiarize participants with the data and facts, the participants first practiced the presentation once without narration. Then we asked them to tell the story to the experimenter with narration. This presentation phase took 40–65 minutes.

Finally, we asked participants to fill out a post-experiment questionnaire about their experience with SketchStory. We also asked a few open-ended follow-up questions about their experience. Sessions lasted an average of an hour with a maximum of 1.5 hours. We captured video and audio of the participants presenting the story.

Results and Discussion

Overall, participants liked SketchStory as a way of telling stories with data, and found it to be easy to learn and use. In a 7-point Likert scale, with 1 = strongly disliked and 7 = strongly liked, the average rating was 5.5. Participants also rated it 5.2 for ease of learning (1 = very difficult to learn and 7 = very easy to learn) and 4.7 for ease of use (1 = very difficult to use and 7 = very easy to use). In particular, participants liked the notion of using a data-enabled canvas instead of sequential slides. The underlying data and synthesis techniques made it easy for them to create charts and aid the narrative storytelling. For example, P3 said "This is my canvas, and it is preloaded with my data, and I can create charts and interact with them with a few strokes, which is very helpful for real time storytelling." P3 also acknowledged the ease of content creation: "Having the dataset embedded with the tool made me comfortable destroying elements as they were no longer needed because I can create them so easily later. When something takes a long time to create, you are not comfortable removing it from the canvas."

Participants also liked the interactivity and connection between the data charts created with sketching. P3 commented, "I like the fact that the visual elements are connected to each other. Interacting with the maps affects other charts in the canvas." P2, a UX researcher with a design background, explained the benefits of interactivity by comparing SketchStory with Photoshop; "I spend so much time in Photoshop or Illustrator meticulously drawing my graphs but I can't go back and change the graph if someone asks a question. But, here my drawings can change according to their questions."

The freeform aspect of sketching facilitated the creation of expressive and personalized data representation. For example, P5 commented that "The iconic data representation helps me to connect with the viewers." Participants also liked the organic graphic style and visual feel of the data charts.

Improvements based on the Usability Issues

Participants found it challenging to create content and perform narration simultaneously, especially the first time. Four participants suggested some preparation beforehand would facilitate the storytelling by reducing real-time sketching burden and cognitive load (and stress) during storytelling. Specifically, one participant (P3) pointed out that the nuances (i.e., heuristics) used for chart generation were confusing and he found himself unsure what chart the system would generate from his sketches. We observed that this was mainly due to the three modal steps for chart creation. Participants often forgot to switch to appropriate mode while sketching example icons and axis baselines. To address this problem, we removed the manual mode switching; we enabled the presenter to record the sequence of charts before the presentation, and to invoke them with the simple sketch gestures during the presentation.

Three participants expressed concerns about their sketching quality in real-time and wanted the system to beautify their sketches. Two participants were concerned that sketching in real-time would create cognitive load and opined that they would like to record the icons for data charts before the presentation. Therefore, we also enabled the presenter to pre-record the example icon as part of the chart specification. Sketch beautification remains future work.

Finally, two participants indicated that annotations should move with the data charts during move operations. As described above, in the new version of SketchStory, the annotations drawn within the chart boundary move along with the chart.

6.5.2 Controlled Experiment

Our goal was to examine the subjective level of engagement of SketchStory for both audience and presenters as compared to a traditional presentation system, PowerPoint. In addition, we wanted to explore how well presenters could learn and use SketchStory.

Datasets and Stories

To compare two systems, we prepared two stories with two datasets [33][31] downloaded from the web. One is the story about global energy consumption we used for the first usability study described earlier. The other story is about global income statistics between 1985 and 2010. To ensure that both stories had comparable length, structure, and complexity, we extracted parts of the datasets and tweaked some numbers. To avoid participants using prior knowledge, we also created two fake region maps and replaced the region names with fictitious country names. For example, Figure 6c shows a region map used for the energy consumption story. Both datasets consist of 5 data columns (country, year, population, energy consumption/gross national income, and per-person consumption/GDP per-capita), with 30 rows in total (5 regions, 6 time points per country).

Both stories had six key messages to convey. For the income statistics story, we had the following six key points:

- Gross national income has increased overall.
- Global population has increased steadily and linearly.
- GDP per-capita has also steadily increased overall.
- Celtica and Aslan are the two countries with most gross income.
- While Celtica's population is more than double of North America's, Aslan's GDP is more than double of Celtica's.
- Celtica and Aslan's national income was once flipped in 2000.

For the SketchStory condition, we used three chart types—bar chart, line chart, and map—for four charts with annotation, filtering, and zooming capabilities. We also recorded an example icon for one chart; presenters had to draw two example icons during their presentations. We suggested possible annotations but did not force presenters to use them. For the PowerPoint condition, we embedded standard charts created with Microsoft Excel; each slide contained one chart except for one page comparing and contrasting the difference between two charts (for the key point #5). Because PowerPoint and Excel do not support maps by default, we instead used a pie chart, and greyed out slices of the pie chart right before drilling down into two countries to help audience follow the transition in the PowerPoint condition (**Figure 6-7**).

Participants, Study Design, and Equipment

We recruited 24 (14 males, 10 females) audience participants. The average age was 38.7, ranging from 28 to 47 years of age. Since they needed to listen to a presentation in English, we required audience participants to be fluent in English. In addition, to examine presenter's reaction to SketchStory, we recruited 5 (4 males, 1 female) presenter participants instead of having a presenter who could have been well prepared for both presentations. The average age was 34.6, ranging from 31 to 43 years of age. Since they needed to give a presentation in English, we required presenter participants to be native English speakers and comfortable with giving presentations. Furthermore, they were screened to be already familiar with PowerPoint and to give presentations regularly (at least once a month). For both audience and presenter participants, we screened them for color-blindness and deafness, and required normal or corrected-to-normal eyesight. They were also required to be able to read basic charts such as line chart, bar chart, and pie chart. Participants were given a software gratuity for their participation.

We conducted the study as a within-subjects design in a small conference room. Each presenter gave a presentation with both systems—SketchStory and PowerPoint—to an audience; the audience watched both presentations as a group. The audience size was four or five participants. To avoid an ordering effect, we counterbalanced the order of the two systems, but we fixed the order of stories. To measure the level of engagement, we collected participants' subjective ratings for each system.

Each presenter gave both presentations on an Intel Xeon W3550 3.07 GHz Windows 8 desktop machine with 12 GB RAM and a 55" Perceptive Pixel Display [30] that supports both pen and touch interaction at a resolution of 1920x1080; both presentations were not projected onto the wall.

Procedure

Presenter participants arrived an hour earlier than the audience participants because they needed to learn the SketchStory system as well as both stories. At the start of each session, we explained the goal and overall procedural of the study and how the dataset was prepared. We then asked presenters to fill out a pre-experiment questionnaire to collect their background about presentation skills.

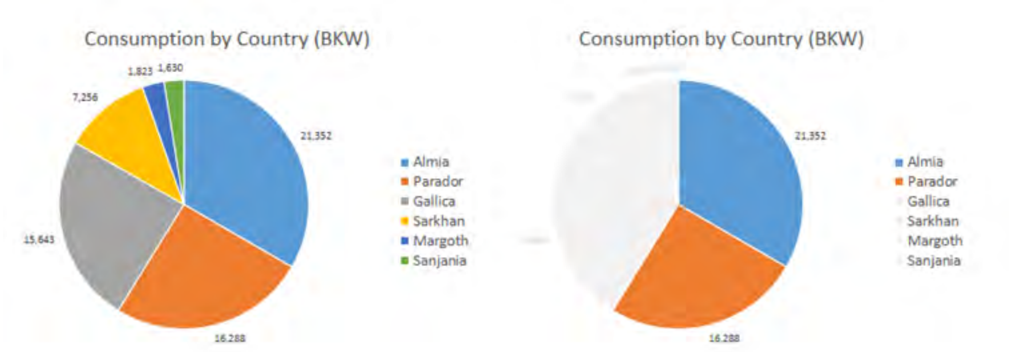


Figure 6-7: For the PowerPoint condition, we greyed out slices of the pie chart right before drilling down into two countries to help audience follow the transition

We then trained the presenters for two stories. As for the story, we explained a storyline and the six key messages to convey during their presentations. We gave the presenters printout notes that had the sequence of screenshots or slides with the key messages, and allowed them to write down their own notes. As for the system training, all presenters were already familiar with PowerPoint, we trained them only on SketchStory. However, for the PowerPoint condition, the presenters practiced the touch swipe gestures because they were not familiar with the touch-enabled large screen. They practiced both presentations multiple times, spending about 30~40 minutes for SketchStory and about 10 minutes for PowerPoint.

When the audience arrived an hour later, we first explained the goal of the study and how the dataset was prepared to the audience in a separate room. We then brought the audience to the conference room where the presenter was waiting. Before the first presentation, we emphasized that we wanted to evaluate the system, not the presenter, even though it might be difficult to separate the two.

Then the presenters told the first story to the audience. After the presenter finished the story, both the audience and presenters were given a short questionnaire about the presentation they just gave and watched, respectively. The same procedure was repeated with the second story and system. On average, the presenters spent about 3 minutes for the SketchStory presentation and about 2 minutes for the

PowerPoint presentation.

At the end of the session, we asked both the audience and presenters to select which system they preferred overall and explain why. In addition, we asked presenter participants to select which system required more efforts and explain why. We captured video and audio of the session. The experiment took about one hour for the audience participants and two hours for the presenter participants.

Results and Discussion: Audience

After each presentation, audience answered four questions intended to measure the subjective level of engagement using a 7-point Likert scale, with 1 = Strongly Disagree and 7 = Strongly Agree for Q1 (enjoyment) and Q4 (perception about presenter's enjoyment), and with 1 = Not engaged at all and 7 = Highly engaged for Q2 (engagement with the story) and Q3 (engagement with the system). Figure 6-8 shows the average subject responses from the audience for both systems. We analyzed these subjective responses using Friedman Chi-Square tests, and found significant differences in all four questions. Audience indicated that they enjoyed the presentation more with SketchStory than PowerPoint ($\chi2(1, N = 24) = 14.73$, $p < .001$), and they felt the presenter enjoyed giving the presentation more with SketchStory than PowerPoint ($\chi2(1, N = 23) = 8.90$, $p = .003$). In addition, they were more engaged with the story ($\chi2(1, N = 24) = 6.37$, $p = .012$) and system ($\chi2(1, N = 24) = 10.89$, $p = .001$) with SketchStory than PowerPoint.

In addition, 75% (18 out of 24) of our audience participants chose SketchStory as their preferred presentation system. When asked why, their reasons were: more engaging, interactive, dynamic, better storytelling, more organic, more personal, annotation, and one screen. Their comments upheld our design rationale to achieve more engaging presentations. For example, P10's comment touches many of these: "I liked how he could draw a picture to have the graph present itself. The view of all the data presented on one screen, options to choose to focus on a particular piece of data. Being able to add text as you go—specifically highlighting what is being discussed.

It's much more interactive and able to keep my attention more interesting than a static presentation."

More specifically, P2's comment demonstrates that SketchStory's real-time content creation successfully created anticipation; "The system makes you want to see what was next. I felt it kept your attention better." In addition, P24's comment shows that interactivity on a canvas helped people follow the story; "The ability to interact with the data and the more organic presentation made it easy to follow the story. Having the entire presentation on one screen allowed greater context in understanding the story."

Interestingly, several audience also commented about presenters' perspective. For example, P7 stated that "It allowed the presenter opportunities to engage his audience, not just present to them." Similarly, P19 mentioned that "It also felt like he [the presenter] was more involved in the presentation, rather than just tossing some dry facts out there." P11 stated his desire to try SketchStory; "I enjoyed watching the presentation, and I would be eager to give a presentation using this system [SketchStory]."

It seems that low-quality sketching can cause distraction, hindering the audience from being more engaged. Six audience participants chose PowerPoint as a preferred system mainly because of the readability. They mentioned that PowerPoint was "easier to read," "legible," and "easier to see and read." Another reason by two audience participants was familiarity. For example, P14 stated that "The ease of visibility understanding the flow, the cleanness of the graphics in each slide the boldness of the fonts much easier to follow along with it as I was more familiar with it."

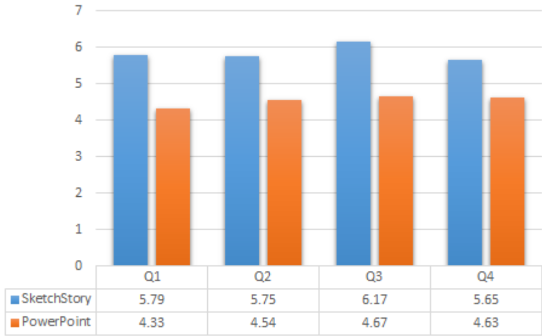


Figure 6-8: Average subjective responses from the audience to 7-point Liker scale questions; 1=Strongly Disagree or Not engaged at all and 7=Strongly Agree or Highly engaged. All differences are significant

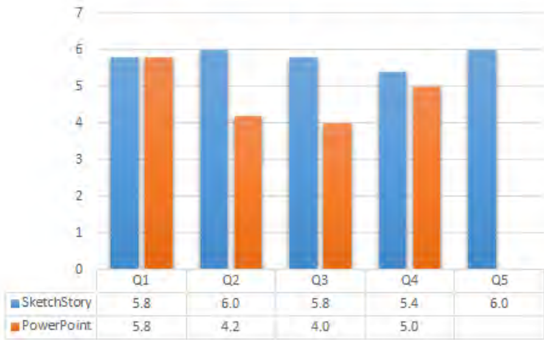


Figure 6-9: Average subjective responses from presenters to 7-point Liker scale questions; 1=Strongly Disagree or Not engaged at all and 7=Strongly agree or Highly engaged

Results and Discussion: Presenter

After each presentation, presenters answered five questions measuring subjective ease of learning, ease of use, and engagement; they answered only four questions for PowerPoint because we did not ask about ease of learning. We again used a 7-point Likert scale, with 1 = Strongly Disagree and 7 = Strongly Agree. **Figure 6-9** shows the average subject responses from presenters for both systems. Since we did not have enough presenters for statistical analysis, we report only descriptive statistics. For SketchStory, Presenters tended to agree with the statement “It was easy to learn” (Q5: 6.0 average; higher than 5.2 from the first usability study). For both SketchStory and PowerPoint, they also tended to agree with the statement "It was easy to give a presentation with this system" (Q1: 5.8 average; higher than 4.7 for ease of use for SketchStory from the first usability study). Presenters' reaction was similar for Q4: "I am satisfied with my presentation with this system." However, SketchStory tended to be ranked higher for Q2 (fun) and Q3 (perception of audience engagement).

Four out of five presenters chose SketchStory as their preferred presentation system even though all five presenters indicated that SketchStory required more effort. They seem to think that SketchStory helped keep the audience more engaged. For example, PP1 stated that "It felt as though I was bringing the audience with me on a journey versus 'presenting to (at)' them." PP5 mentioned that "It is almost like I'm telling a story rather than stating facts, which is more fun/entertaining/informative/memorable for everyone involved." Not surprisingly, the most common reason given for more effort was the fact that they had to draw the icons to populate the graphs. Two presenters also commented that, for SketchStory, they had to be more familiar with the content.

6.6 Discussion and Future Work

The encouraging results of the controlled study indicate that the SketchStory design—melding expressiveness, dynamic sketch, and interactivity—supports more engaging storytelling. The results from both studies also point out the unique affordances and potential of the approach, and provide exciting possibilities in the direction of novel and engaging storytelling with data.

Integration with Exploration: SketchStory was designed for storytelling and presentation, assuming that the presenter had already found a story (a sequence of data charts) to tell. However, participants spontaneously commented about the capabilities of data-bound sketching for data exploration. For example, two participants in the first study mentioned that they would like to have the capability of changing the chart type dynamically (e.g., from bar chart to line chart). We have extended SketchStory to support chart type changes for the charts with x and y axes through simple sketch gestures. For example, you can draw a bar and an 'M' shape in the chart area to switch to a bar chart and line chart, respectively. However, an example icon good for one chart type is not necessarily right for the other chart type. For example, thin-and-tall icons are good for bar charts but not so great for line charts. Therefore, we might need to further extend SketchStory to support multiple example icons for different chart types.

As explained in the Related Work section, there has recently been research on leveraging sketch-based interaction for fluid data exploration. To ease the problem of mode switching, SketchStory enables the presenter to record a sequence of charts before the presentation, and to invoke them with simple sketch gestures during the presentation. However, there is still room for improvement in terms of authoring the sequence of charts. Incorporating exploration capabilities will benefit SketchStory by enabling easy sequence authoring. Therefore, we have been expanding SketchStory accordingly to provide people with a seamless experience from fluid data exploration to engaging presentation.

Burden vs. Control: Interestingly, most presenter participants preferred SketchStory even though it required more work. This means that lowering perceived workload does not always lead to the optimal experience. We suspect this is because of the trade-off between burden and control. For example, we could minimize the burden during the presentation by allowing presenters to record additional chart settings such as the size and location of the charts, and then to invoke them with simple click-through. However, presenters then lose an additional measure of control during the presentation, and the experience approaches that of the traditional presentation. It is important to note that some presenter participants liked having the control and believed that it helped them keep the audience more engaged.

In the current implementation, the presenter can see the chart sequence in a dropdown list and change the order at any time by selecting a different thumbnail.

However, for fluid presentations, presenters need to remember the sequence without interrupting their flow. Existing commercial presentation tools like PowerPoint and Keynote provide a presenter view mode, where the presenter could preview the next slide along with a current slide on a computer screen. It would be useful to explore how to help presenters better remember or recall the flow.

Sketch Rendering: During the implementation of tally chart rendering for initial SketchStory, we had a trade-off between data precision and aesthetics. For instance, in **Figure 6-10**, data values of 307 and 365 have an equal number (five) of icons. We originally decided not to clip or distort icons for three reasons. First of all, we wanted to provide more aesthetically pleasing drawings; clipped icons looked untidy. We identified similar practices by infographic designers during our analysis phase, preferring aesthetics to precision (e.g., **Figure 6-2**a). Ultimately, our main goal was to support storytelling scenarios that demonstrate major trends without depending on exact precision. Second, given the imprecise nature of sketchiness, we believed that people would be more forgiving of imprecision in sketched content. In fact, during our first usability study, participants did not complain about imprecision. Rather, they used annotations to point out major trends. However, how the sketchy and iconic rendering of data charts affect the perceived accuracy requires further investigation.

Regarding the sketchy rendering style, one participant commented that SketchStory would be more useful for less business-based presentation, such as art or gaming. People may infer a sketchy style to be more fun and entertaining rather than official and serious. It would be interesting to investigate the possible connotations for sketchy rendering more formally in future work.

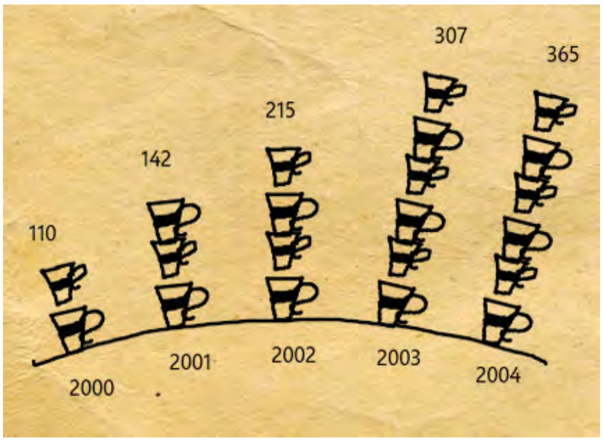


Figure 6-10: Example tally chart generated by initial SketchStory

In contrast to the first usability study, the presenter participants for the controlled experiment did not express concerns about sketch qualities. This might be because the participants we recruited for the first study had more design and sketch experience. Or it could be because SketchStory now enables presenters to record before the presentation; presenters did not have to draw the most complex icons during their presentation. Either way, some audience participants preferred PowerPoint because SketchStory looked untidy and cluttered for them. Furthermore, one major issue with SketchStory for the audience was readability; each chart did not have a title like a PowerPoint slide and the fonts for the numbers and labels were too small. An audience participant mentioned that it would be nice to have the capability of maximizing one chart since it would alleviate the readability problem. It could be helpful to incorporate existing sketch beautification algorithms and to support adaptive font size.

Enhancing SketchStory: Participants indicated additional straightforward features that could improve the user experience. While SketchStory was designed and developed for real-time storytelling with data, presenter participants expressed their desire to distribute the story and ideas asynchronously, in the form of video so that people can watch the story offline without a presenter.

Often times telling a story with data is a part of a more general presentation, and the current SketchStory design is not sufficient for general presentations. There are two ways to combine SketchStory with a general presentation tool such as PowerPoint, without relying on application switch during the presentation. First is to embed SketchStory as a plug-in into the general presentation tool. Second is to integrate more traditional slide content into SketchStory including multimedia content (i.e., images, video, etc.), which will facilitate richer and more general storytelling capabilities.

Participants also acknowledged that SketchStory's capability of digitizing whiteboards could be useful in other domains. For example, they mentioned that the SketchStory approach would be great for lectures and group meetings because it allows for dynamic group interaction. In the controlled experiment, we mainly investigated the subjective level of engagement for stories with data. It would be useful to investigate the effectiveness of the SketchStory approach for content understanding or idea generation, not just in data story presentations but also in other contexts such as education.

6.7 Summary

Inspired by successful storytelling techniques, we explored a novel approach to telling stories with data by melding the expressive visual language of infographics with the narrative storytelling attributes of whiteboard animation. SketchStory is an interactive whiteboard system integrating real-time freeform sketching capabilities with the fluid synthesis of interactive, organic data-bound charts. We first conducted a preliminary usability study to understand how people would use this new form of storytelling. We improved the system based on the lessons learned from the first study. For example, to reduce the burden of manual manipulation during the presentation, SketchStory allows the presenter to record a sequence of charts before the presentation, and invoke them with simple sketch gestures in real-time. We then conducted a controlled experiment to compare SketchStory with Microsoft PowerPoint, one of the most commonly used presentation tools. Results show that the audience is more engaged with the presentation done with SketchStory than PowerPoint, and that most presenters favored SketchStory even though they acknowledged the extra effort required to present with SketchStory. In addition, our results provide insights for new possibilities for future work in sketch-based narrative storytelling with data.

While the design of SketchStory seeks inspiration from multiple media, it bears a similar design process like the previous tools, in contrast to previous projects presented in this thesis. For the process, SketchStory seeks inspiration from whiteboard animation, where presenter sketches along with synchronized narration to vividly tell a story. The expressive graphic style makes whiteboard animation a very unique and engaging form of storytelling. The real-time dynamic sketching conveys the order of action sequences by directing viewer attention from one object to the next, building anticipation as in a story, making it an effective storytelling technique to plan ideas, emotions and thoughts. The inspiration for the artifacts comes from Infographics- expressive graphical representations of information, data, or knowledge. Infographics communicate complex information in an aesthetically pleasing way, employing icons and other visual elements that are customized to the dataset. Infographics allows presenters the artistic freedom to create a unique, personalized chart taking full advantage of an innate visual language that is largely universal. Finally, as a medium, The main goal of storytelling with data is to communicate the key patterns and observations from data exploration phase effectively. One key desired attribute of this medium is interactivity, inviting verification, further questioning, and exploration of alternative explanations. As such, SketchStory seeks inspiration from interactive information visualization in this aspect.

7. CONCLUSION

7.1 Contribution

This thesis presents a series of new media digital arts and communication tools inspired by traditional art media.

SandCanvas is a digital multi-touch application for real-time storytelling inspired by sand animation. Sand animation is a form of visual storytelling in which an artist dexterously manipulates fine granules of sand to produce images and animations. Our analysis of traditional sand animation pointed to the fact that the process of creating those visuals consists of a stream of powerful and expressive hand gestures, leveraging the delicate mechanical structure and physical affordances of human hands to create a wide range of visual effects. ***SandCanvas*** uses a computer vision technique based upon the principles of diffused illumination, thus preserving the expressiveness of human hand gestures. This algorithm strikes a balance between realism and performance. The user study of ***SandCanvas*** demonstrates the variety of hand gestures at artists' disposal.

Vignette is a style-preserving sketching tool for pen-and-ink illustration with built-in texture synthesis capabilities. Pen-and-ink illustration is a popular art medium, incorporating a wealth of artistic styles and textures, but manually rendering pen-and-ink illustrations takes an inordinate amount of time and skill. Existing texture illustration tools are either automatic, losing artistic style, or they rely on tweaking numerous parameters, destroying artists' workflow. Our analysis of traditional ink illustration artifacts suggested a cluster of texture filling techniques: brush, fill, and hatch. The design of ***Vignette*** preserves artists' traditional workflow with freeform sketching and direct manipulation. The underlying texture synthesis techniques reduce tedium but preserve artistic styles, since textures are generated from examples provided by the users.

Draco extends the spatial synthesis capabilities of ***Vignette*** into the temporal domain. As a medium, ***Draco*** takes inspiration from Cinemagraphs, falling somewhere in between videos and static pictures. Illustrations with Draco capture the living qualities of a moment with continuous dynamic phenomena, yet exhibit the unique, timeless nature of a still picture. In this project, we limit our focus to groups of objects with coordinated motions, which characterizes a variety of phenomena around us. One key insight to animating groups of objects is that their motion control can be divided into coarse and granular scales. The design of ***Draco*** is based upon kinetic texture, an animation framework providing coarse-to-fine scale motion controls with freeform sketching and direct manipulation. The design of ***Draco*** simultaneously achieves generality, controllability, and ease of use.

SketchStory is a tool for telling stories with data through freeform sketching on interactive whiteboards. Presenting and communicating insights to an audience – telling a story – is one of the main goals of data exploration. Even though visualization as a storytelling medium has recently begun to gain attention, storytelling is still underexplored in information visualization and little research has been done to help people tell their stories with data. To create a new, more engaging form of storytelling with data, ***SketchStory*** leverages and extends the narrative storytelling attributes of whiteboard animation with pen and touch interactions. The design of ***SketchStory*** seeks insight from whiteboard animation, infographics, and interactive visualization. ***SketchStory***, a data-enabled digital whiteboard, facilitates the creation of personalized and expressive data charts quickly and easily. ***SketchStory*** recognizes a small set of sketch gestures for chart invocation and automatically completes charts by synthesizing the visuals from the presenter-provided example icon and binding them to the underlying data. Results show that the audience is more engaged by presentations done with ***SketchStory*** than PowerPoint.

In summary, this thesis presents four digital art and communication tools for creative self-expression and content creation, along with the following design insights:

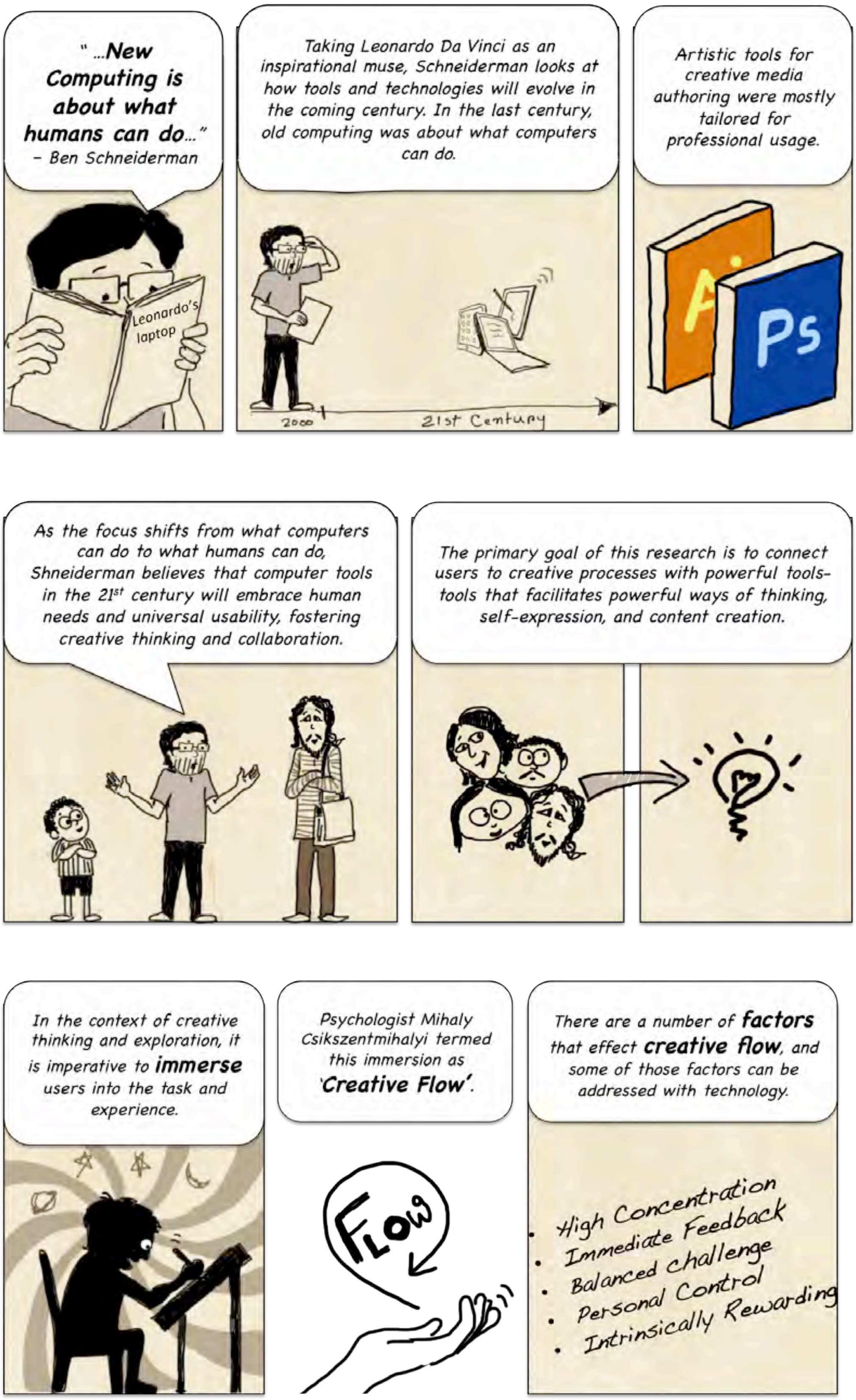
- These tools transform existing physical art media to a digital platform that not only preserves the graphical style of traditional art but also preserves physical workflow and interaction style.
- While preserving the style of the artist, art form, and essential workflow of the creation process, these tools use the capabilities offered by digital technology to accelerate the tedious components of the original process by synthesizing from example sketches spatially (***Vignette***), spatio-temporally (***Draco***), or from underlying data (***SketchStory***).
- The design of these tools capitalizes on the freeform nature of sketching, gestural interaction, and direct manipulation, in contrast to indirect parameter tweaking, coding, and scripting for digital content creation.
- The design process does not simply replicate an existing physical art medium into the digital domain. Rather, the new affordances of the digital domain (editing, archival and rapid exploration) re-form the original medium into a new one. ***SandCanvas's*** gesture recording/playback and mixed media features capability goes beyond traditional sand animation. ***Vignette*** and ***Draco*** facilitate animation authoring and rapid exploration, which is difficult to do with traditional ink illustration and animation tools. The timeless nature of the animated illustrations of Draco makes it a new medium by itself, placing it in between video and still pictures. ***SketchStory*** seeks inspiration from multiple existing media and combines the unique affordances of those media to create a new one for storytelling with data in real-time.
- User studies with professionals and amateurs indicate the expressiveness, unique affordances, and creative possibilities of these tools. After training sessions, even first-time users were able to create artifacts within minutes, pointing to the ease of learning and usage of the interfaces. The ability to create contents inspired exploration and creativity. The resulting artifacts captured the artists' personal

styles. In general, while novice and amateur users were thrilled with the capabilities of the tools, professional artists asked for additional features and advanced capabilities for production quality artifacts.

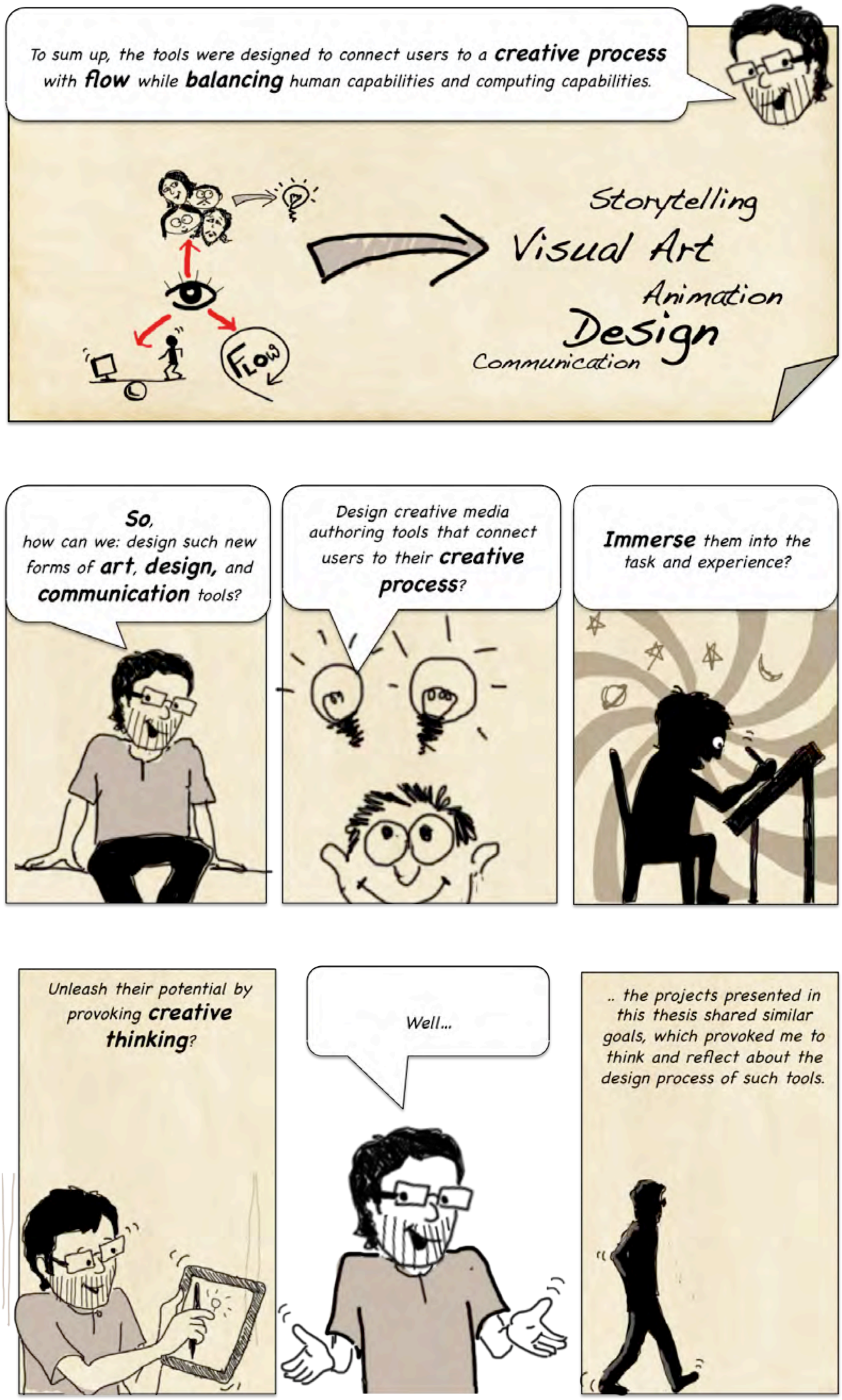
To design new forms of digital art and communication tools, it is crucial to understand the role of humans and computers in creative tasks and how they work best with each other by combining their complementary strengths. The tools presented in this thesis employ powerful end-user-programming capabilities by taking sample input from the users and performing repetitive tasks where necessary. This approach reduces the tedium of repetitive tasks, yet preserves the artistic styles and expressiveness.

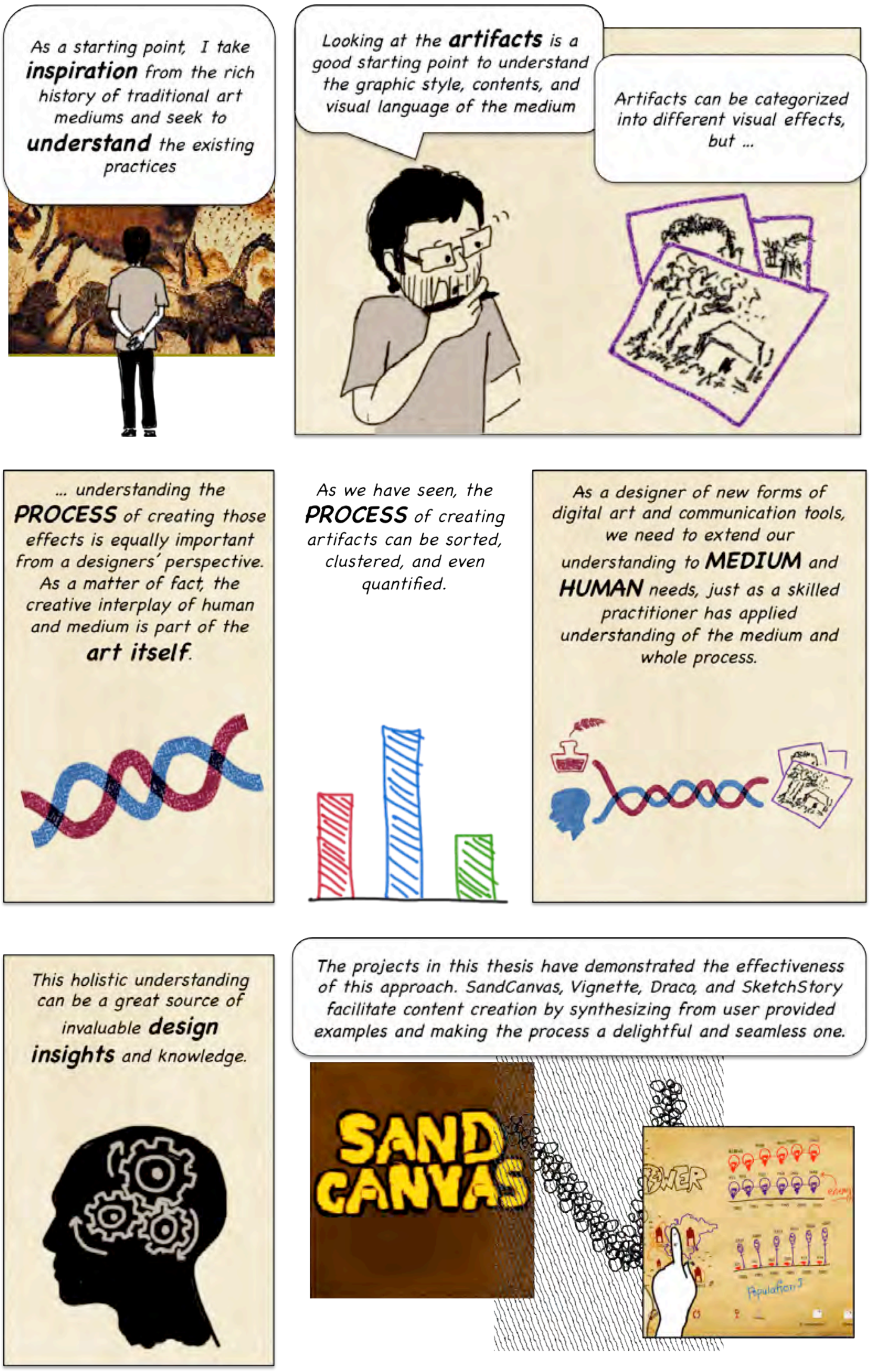
Furthermore, our ancestors developed sophisticated art media, tools, and techniques over thousands of years. Traditional art media, such as oil painting, pen illustrations, and sand art, capitalize on the physical affordances of artists and the tangible nature of their tools. In my design process for these tools, I studied different components of traditional art practice to gain design insights and inspiration. In traditional art media, an artist interacts with the tools and materials and goes through a process to create the final artifacts. There is a plethora of physical art media with distinct raw materials, visual styles, affordances, and creation process. Artifacts, the final drawings and artworks, aid the understanding of visual style, grammar, and associated parameters of the chosen medium. The process of traditional workflow helped me to understand how an artist translates visual form (or thoughts) into structures and the steps to create final artifacts with tools. While the components of traditional art can be a great source of design insights and inspiration, the design of new digital art and communication tools should combine the best of traditional art practices and digital affordances, enabling the creation of entirely new forms of expression.

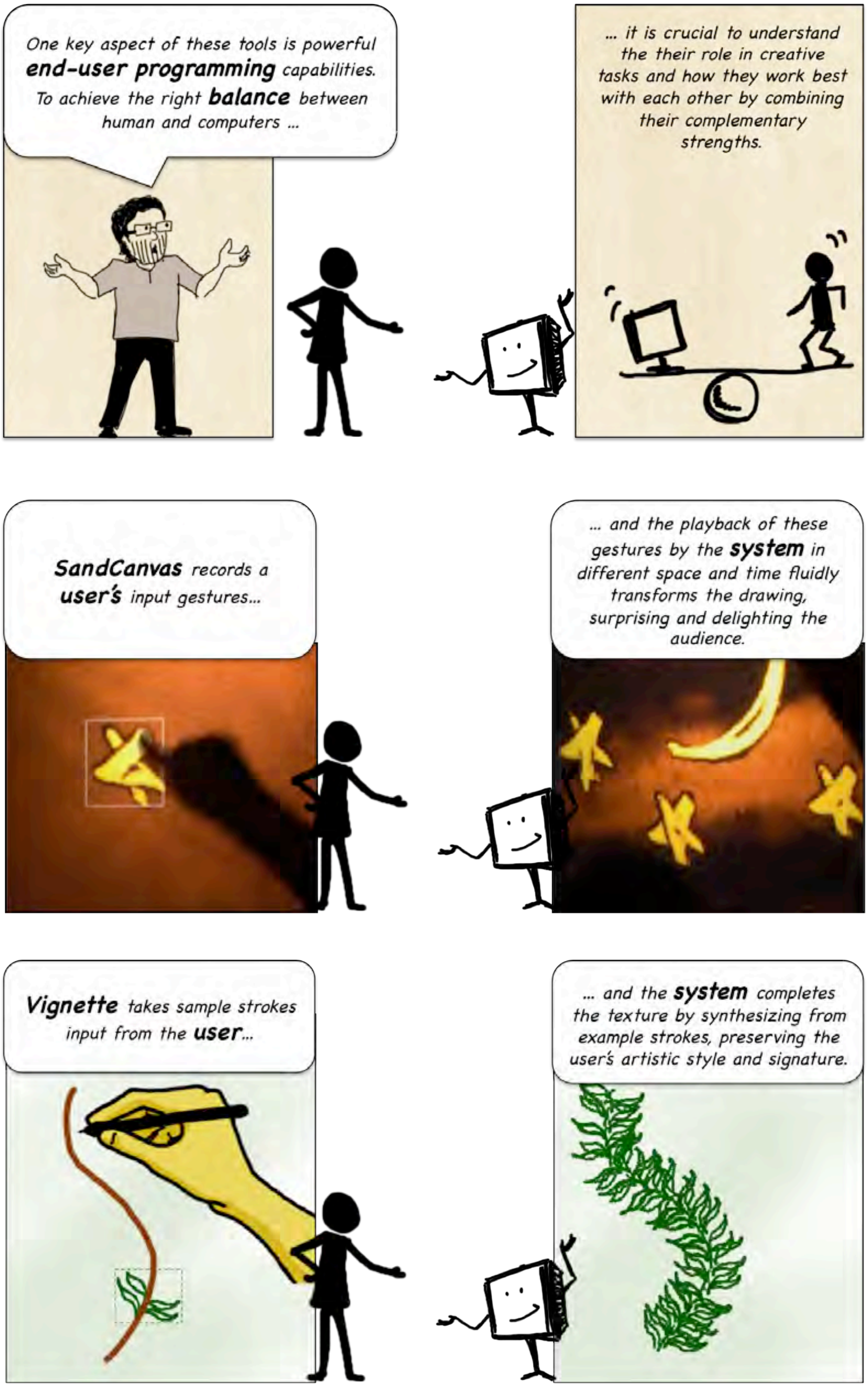
7.2 Summary and Future works




Careful design of user experience (UX) can facilitate immersion...
UX
... and creative flow.
FLOW
In the digital art domain - it's crucial to understand the complimentary abilities between the human and the computer.
Humans are good at reasoning, interpretation, and creative thinking...
... while computers are good at numbers, calculation, and multiplicity - performing repetitive tasks.
In the context of creativity support tools, it is crucial to understand how these complementary affordances can work best with each other.
Tools that provide starting templates or provide aesthetic suggestions make it easy for novices, but at the cost of reduced artistic freedom... losing expressiveness, and originality.
Tools (e.g., MS Paint) that thoughtlessly do whatever a human commands with buttons and menus can be tedious to use ... destroying the flow and experience.
Designing 'smart' tools (for example, with end-user programming) with the right balance between human and computer is the best way to extend human capabilities for creative expression.


To sum up, the tools were designed to connect users to a **creative process** with **flow** while **balancing** human capabilities and computing capabilities.
FLOW
Storytelling
Visual Art
Animation
Design
Communication
So, how can we: design such new forms of **art**, **design,** and **communication** tools?
Design creative media authoring tools that connect users to their **creative process**?
Immerse them into the task and experience?
Unleash their potential by provoking **creative thinking**?
Well...
.. the projects presented in this thesis shared similar goals, which provoked me to think and reflect about the design process of such tools.


As a starting point, I take inspiration from the rich history of traditional art mediums and seek to understand the existing practices
Looking at the artifacts is a good starting point to understand the graphic style, contents, and visual language of the medium
Artifacts can be categorized into different visual effects, but ...
... understanding the PROCESS of creating those effects is equally important from a designers' perspective. As a matter of fact, the creative interplay of human and medium is part of the art itself.
As we have seen, the PROCESS of creating artifacts can be sorted, clustered, and even quantified.
As a designer of new forms of digital art and communication tools, we need to extend our understanding to MEDIUM and HUMAN needs, just as a skilled practitioner has applied understanding of the medium and whole process.
This holistic understanding can be a great source of invaluable design insights and knowledge.
The projects in this thesis have demonstrated the effectiveness of this approach. SandCanvas, Vignette, Draco, and SketchStory facilitate content creation by synthesizing from user provided examples and making the process a delightful and seamless one.
SAND CANVAS


One key aspect of these tools is powerful end-user programming capabilities. To achieve the right balance between human and computers ...
... it is crucial to understand the their role in creative tasks and how they work best with each other by combining their complementary strengths.
SandCanvas records a user's input gestures...
... and the playback of these gestures by the system in different space and time fluidly transforms the drawing, surprising and delighting the audience.
Vignette takes sample strokes input from the user...
... and the system completes the texture by synthesizing from example strokes, preserving the user's artistic style and signature.


Draco takes sample sketches (for instance, raindrops) from the user...
... and the system creates animated rainfall by spatio-temporal synthesizing from example strokes.
SketchStory takes sketched icons as input from the user....
... and the system generates sketchy rendering charts by synthesizing example icon with underlying data.
While these interfaces facilitate content creation by synthesizing from user provided examples, the resulting contents accommodate a wide range of artistic styles and personal signatures.


Incorporating **traditional workflow** in these interfaces with pen-based interaction and direct manipulation facilitates creative flow, learning, and content creation – even for amateurs.
Symmetrical Skinny Pouring
Symmetrical Hand Sweep
Designing new forms of art and communication tools requires creative interdisciplinary thinking and holistic understanding.
As PIXAR's **John Lasseter** once stated,
Art challenges technology, Technology inspires art
The success of PIXAR is a shining example of how the **intersection** of artistic vision and technological excellence can create a magical experience and push the limits of both art and technology.
Science
Art
It's the **right time** for broadening the horizons of media authoring capabilities to the creative hands of mass users. The projects in this thesis are just the beginning for me. In the future, I intend to develop new tools for content creation and explore **design research** of such tools in depth.
... THE END ...

PUBLICATIONS

Paper/Journal

1. Draco: Bringing Life to Illustrations with Kinetic Textures
   R H Kazi, F Chevalier, T Grossman, S Zhao, G Fitzmaurice
   CHI 2014 (10 pages) ***Honorable Mention Award; Golden Mouse Award; People's choice Best Talk Award***

2. SketchStory: Telling more Engaging Stories with Data Through Freeform Sketching
   B Lee, R H Kazi, G Smith
   Infovis 2013 (10 pages) ***Appeared in Microsoft Techfest 2013 Keynote Speech***

3. Vignette: Interactive Texture Design and Manipulation with Freeform Gestures for Pen-and-Ink Illustration
   R H Kazi, T Igarashi, S Zhao, R C Davis
   CHI 2012 (10 pages) ***Invited to CHI 2012 Interactivity***

4. SandCanvas: A Multi-touch Art Medium Inspired by Sand Animation
   R H Kazi, K C Chua, S Zhao, R C Davis, and K L Low
   CHI 2011 (10 pages) ***Honorable Mention Award; Golden Mouse Award***

Conference Proceedings

1. Draco: Living Illustrations
   R H Kazi, F Chevalier, T Grossman, S Zhao, G Fitzmaurice
   SIGGRAPH 2014 *Studio and Poster*

2. Draco: Living Illustrations
   R H Kazi, F Chevalier, T Grossman, S Zhao, G Fitzmaurice
   CHI 2014 *Interactivity and Video Showcase* ***ACM Golden Mouse Award***

3. Digital Arts for End-Users: Learning from Traditional Craft Practice
   R H Kazi
   CHI 2013 *Doctoral Consortium*

4. Vignette: Style Preserving Sketching Tool for Pen-and-Ink Illustration with Texture Synthesis
   R H Kazi, T Igarashi, S Zhao, R C Davis, K Takayama
   SIGGRAPH 2012 *Poster and Studio Talk*

5. Anyone Can Sketch Vignettes
R H Kazi, T Igarashi, S Zhao, R C Davis, TJ Montessat
CHI 2012 *Video Showcase*

6. SandCanvas: New Possibilities in Sand Animation
R H Kazi, K C Chua, S Zhao, R C Davis, and K L Low
SIGGRAPH 2011 *Dailies and Studio Talks*

7. SandCanvas: New Possibilities in Sand Animation
R H Kazi, K C Chua, S Zhao, R C Davis, and K L Low
CHI 2011 *Video Showcase* ***ACM Golden Mouse Award for 'Best design'***

REFERENCES

1. (Ed.) CM-T, (2010). Tabletops - Horizontal Interactive Displays. Springe.
2. A.M. Paul. (1912). Your Brain on Fiction. Neuroscience, 9, 19.
3. Adobe. After Effects. tinyurl.com/AdobeAfterEffectsCC
4. Adobe. Flash. www.adobe.com/products/flash.html
5. Autodesk. 3D Studio Max. tinyurl.com/Adsk-3Dsmax
6. Autodesk. Maya. tinyurl.com/Adsk-maya
7. B. Buxton. (2010). Sketching User Experiences: Getting the Design Right and the Right Design: Getting the Design Right and the Right Design, Morgan Kaufmann Publishers.
8. Bacher HP. (2007). Dream Worlds: Production Design for Animation. Focal Press.
9. Barla P, Breslav S, Thollot J, Sillion F and Markosian L. (2003). Learning Style Translation for the Style of a Drawing. ACM Transactions of Graphics 22,1, 33 - 46
10. Barla, P., Breslav, S., Thollot, J. and Markosian, L. (2006). Interactive hatching and stippling by example. INRIA Technical Report.
11. Barnes C, Jacobs, D. E., Sanders, J., Goldman, D. B., Rusinkiewicz, S., Finkelstein, A., & Agrawala, M. (2008). Video puppetry: a performative interface for cutout animation. In. ACM SIGGRAPH Asia*,* 1-9.
12. Bau, O., Poupyrev, I., Israr, A., & Harrison, C. (2010). TeslaTouch: electrovibration for touch surfaces. In Proceedings of the 23nd annual ACM UIST (pp. 283-292).
13. Baudisch, P., Tan, D., Collomb, M., Robbins, D., Hinckley, K., Agrawala, M., Zhao, S., and Ramos, G. (2006). Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects. ACM UIST. 169-174.
14. Beiman, N. (2012). Prepare to board! Creating story and characters for animation features and shorts. Taylor & Francis.
15. Bell N, Y.Yu, and P.J.Mucha. (2005) Particle Based Simulation of Granular Materials. ACM SIGGRAPH.
16. Besl PJ and McKay ND. (1992). A Method for Registration of 3-D Shapes*.* IEEE Trans. on Pattern Analysis and Machine Intelligence; 14(2): 239-256.
17. Bonham, S. (2007). Graphical response exercises for teaching physics. The Physics Teacher, 45, 482.

18. Buxton, B. (2007). Sketching User Experiences: Getting the Design Right and the Right Design. Morgan Kaufmann Publishers Inc.

19. C. Williamson and B. Shneiderman. (1992). The dynamic HomeFinder: Evaluating dynamic queries in a realestate information exploration system, Proc. Research and Development in Information Retrieval, pp. 338-346.

20. Cavanagh, P. (2005). The artist as neuroscientist. Nature, 434(7031), 301-307.

21. Chen, G., Esch, G., Wonka, P., Müller, P., and Zhang, E. (2008). Interactive procedural street modeling. ACM SIGGRAPH. Art. 103.

22. Cho. J., Xenakis, A., Gronsky, S., and Shah, A. (2007) Course 6: Anyone can cook: inside ratatouille's kitchen. ACM SIGGRAPH Courses.

23. Chuang, Y., Goldman, D., Zheng, K., Curless, B., Salesin, D., and Szeliski, R. (2005). Animating pictures with stochastic motion textures. ACM SIGGRAPH. 853.

24. Cinemagraphs. cinemagraphs.com

25. Cohen, H. (1995). The further exploits of AARON, painter. Stanford Humanities Review, 4(2), 141-158.

26. Cole, F., Golovinskiy, A., Limpaecher, A., Barros, H. S., Finkelstein, A., Funkhouser, T., & Rusinkiewicz, S. (2008). Where do people draw lines?. In ACM Transactions on Graphics (TOG) (Vol. 27, No. 3, p. 88).

27. Cole, F., Sanik, K., DeCarlo, D., Finkelstein, A., Funkhouser, T., Rusinkiewicz, S., & Singh, M. (2009). How well do line drawings depict shape?. ACM Transactions on Graphics-TOG, 28(3), 28.

28. Csikszentmihalyi, M. (1990). Flow: The Psychology of Optimal Experience. (1st edition). New York: Harper and Row.

29. Cutting, J. E. (2002). Representing motion in a static image: constraints and parallels in art, science, and popular culture. PERCEPTION-LONDON, 31(10), 1165-1194.

30. D. Roam. (2008). The back of the napkin (expanded edition): Solving problems and selling ideas with pictures. Penguin Group, Inc.

31. D.A. Keim. (2002). Information visualization and visual data mining, IEEE TVCG, vol. 8, no. 1, pp. 1-8.

32. Danielson P-E. (1980). Euclidean distance mapping. Computer Graphics and Image Processing; 14: 227-248.

33. Data | The World Bank, http://data.worldbank.org.

34. Davis RC, Colwell B, and Landay JA. (2008). K-sketch: a 'kinetic' sketch pad

for novice animators. ACM CHI, 413-422.

35. Davis, R. and Landay, J. (2004). Informal Animation Sketching: Requirements and Design. AAAI. 42-48.

36. DeCarlo, D., & Santella, A. (2002, July). Stylization and abstraction of photographs. In ACM Transactions on Graphics (TOG) (Vol. 21, No. 3, pp. 769-776). ACM.

37. Deussen, O., Hiller, S., Overveld, C. and Strothotte, T. (2000). Floating Points: A Method for Computing Stipple Drawings. IEEE Computer Graphics Forum 19,3, 41 - 50

38. E. Segel and J. Heer. (2010). Narrative visualization: Telling stories with Data. IEEE TVCG (Proc. InfoVis), vol. 16, no. 6, pp. 1139-1148.

39. Efron D, (1941). Gesture and Environment. New York: King's Crown Press.

40. Efros, A.A. and Freeman W.T. (2001). Image quilting for Texture Synthesis and Transfer. In Proc. ACM SIGGRAPH, 341 - 346

41. Efros, A.A. and Leung, T.K. (1999). Texture Synthesis by Non-parametric sampling. IEEE Computer Vision Vol.2, 1033 – 1038

42. Ellis, T. O., Heafner, J. F., & Sibley, W. L. (1969). The GRAIL Project: An experiment in man-machine communications. (No. RM-5999-ARPA). RAND CORP SANTA MONICA CA.

43. F.B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. (2007) Many Eyes: a site for visualization at internet scale. IEEE TVCG (Proc. InfoVis), vol. 13, no. 6, pp. 1121-1128.

44. Fernquist, J., Grossman, T., and Fitzmaurice, G. (2011). Sketch-sketch revolution: an engaging tutorial system for guided sketching and application learning. ACM UIST. 373-382.

45. G. Li, X. Cao, S. Paolantonio, and F. Tian. (2012). SketchComm: a tool to support rich and flexible asynchronous communication of early design ideas. Proc. CSCW, pp. 359-368.

46. Goodwin, T., Vollick, I., & Hertzmann, A. (2007). Isophote distance: a shading approach to artistic stroke thickness. ACM NPAR (pp. 53-62).

47. Gröller, E. and Szirmay-Kalos, L. (2006). Stroke Pattern Analysis and Synthesis. EUROGRAPHICS, 25, Number 3.

48. Guptill A.L. (1976). Rendering with pen and ink. Watsom-Guptill Publications, New York.

49. Hancock M, Cate T, Carpendale S, Isenberg T. (2010). Supporting Sandtray Therapy on an Interactive Tabletop. ACM CHI.

50. Heiser, J., and Tversky, B. (2006). Arrows in comprehending and producing mechanical diagrams. Cognitive Science. 30:581-592.

51. Hertzmann, A. (2010). Non-photorealistic rendering and the science of art. ACM NPAR (pp. 147-157).

52. Hertzmann, A., Oliver, N., Curless, B., Seitz, S.M. (2002). Curve Analogies. In Proc. Rendering Techniques, 233- 246

53. Hinckley K, Baudisch, P., Ramos, G., & Guimbretiere, F. (2005). Design and Analysis of Delimiters for Selection-action Pen Gesture Phrases in Scriboli. ACM CHI, 451-460.

54. Hinckley, K., Chen, X. A., & Benko, H. (2013). Motion and context sensing techniques for pen computing. In Proceedings of the 2013 Graphics Interface Conference (pp. 71-78). Canadian Information Processing Society.

55. Your brain on fiction (2012). New York Times. http://www.nytimes.com/2012/03/18/opinion/sunday/the-neuroscience-of-your-brain-on-fiction.html.

56. I.E. Sutherland. (1963). Sketchpad: A man-machine graphical communication system. Proc. AFIPS Spring Joint Comp. Conf, pp. 329-346.

57. Igarashi T, Moscovich T, and Hughes JF. (2004) As-rigid-as-possible shape manipulation. In. ACM SIGGRAPH 2005 Papers, 1134-1141.

58. Igarashi, T. and Hughes, J.F. (2002). Clothing Manipulation In Proc. ACM UIST, 91- 100.

59. Igarashi, T., Matsuoka, S., & Tanaka, H. (2007). Teddy: a sketching interface for 3D freeform design. In ACM SIGGRAPH 2007 courses (p. 21).

60. Ijiri, T., Mech, R., Miller, G. and Igarashi, T. (2008). An example-based procedural system for element arrangement. Computer Graphics Forum 27,2 (EUROGRAPHICS), 429 – 436.

61. Ishii, H., & Ullmer, B. (1997). Tangible bits: towards seamless interfaces between people, bits and atoms. ACM CHI (pp. 234-241).

62. Islam, M. T., Nahiduzzaman, K. M., Why, Y. P., & Ashraf, G. (2010). Learning character design from experts and laymen. IEEE Cyberworlds (CW), pp. 134-141.

63. J. Browne, B. Lee, S. Carpendale, N. Riche, and T. Sherwood. (2011). Data analysis on interactive whiteboards through sketch-based interaction. Proc. ITS, pp. 154-157.

64. J. Heer, F. Viegas, and M. Wattenberg. (2007). Voyager and voyeurs: Supporting asynchronous collaborative information visualization. ACM CHI, pp. 1029-1038.

65. J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. (2008). Graphical histories for visualization: Supporting analysis, communication, and evaluation. IEEE TVCG (Proc. InfoVis), vol. 14, no. 6, pp. 1189-1196.

66. J. Hsu. (2008). The secrets of storytelling. Scientific American Mind, vol. 19, no. 4, pp. 46-51.

67. J. Landay and B. Myers. (1995). Interactive sketching for the early stages of user interface design. ACM CHI, pp. 43-50.

68. J. Lin, M.W. Newman, J.I. Hong, and J.A. Landay. (2001). DENIM: an informal tool for early stage web site design. Ext. Abs. CHI, pp. 205-206.

69. J. Schumann, T. Strothotte, S. Laser, and A. Raab. (1996). Assessing the effect of non-photorealistic rendered images in CAD. ACM CHI, pp. 35-41.

70. J. Walny, B. Lee, P. Johns, N.H. Riche, and S. Carpendale. (2012) Understanding pen and touch interaction for data exploration on interactive whiteboards. IEEE TVCG (Proc. InfoVis), vol. 18, no. 12, pp. 2779-2788.

71. J. Wood, P. Isenberg, T. Isenberg, J. Dykes, N. Boukhelifa, and A. Slingsby. (2012). Sketchy rendering for information visualization. IEEE TVCG (Proc. InfoVis), vol. 18, no. 12, pp. 2749-2758.

72. J.J. LaViola and R.C. Zeleznik. (2007). MathPad2: A system for the creation and exploration of mathematical sketches. ACM SIGGRAPH 2007 Courses.

73. J.N. Bott and J.J. LaViola Jr. (2010). A pen-based tool for visualizing vector mathematics. Proc. SBIM, pp. 103-110.

74. Johnson, G., Gross, M. D., Hong, J., & Yi-Luen Do, E. (2009). Computational support for sketching in design: a review. Foundations and Trends in Human-Computer Interaction, 2(1), 1-93.

75. Joshi, N., Mehta, S., Drucker, S., Stollnitz, E., Hoppe, H., Uyttendaele, M., and Cohen, M. (2012). Cliplets: juxtaposing still and dynamic imagery. ACM UIST. 251-260.

76. K. Hinckley, K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, and B. Buxton. (2010). Pen + touch = new tools. ACM UIST, pp. 27-36.

77. K. Ryall, N. Lesh, T. Lanning, D. Leigh, H. Miyashita, and S. Makino. (2005). QueryLines: Approximate query for visual browsing. Ext. Abs. CHI, pp. 1765-1768.

78. Kalnins, R.D., Markosian, L., Meier, B.J., Kowalski, M.A., Lee, J.C., Davidson, P.L., Webb, M., Hughes, J.F. and Finkelstein, A. (2002). WYSIWYG NPR: Drawing Strokes Directly on 3D Models. ACM SIGGRAPH 21,3, 755-762

79. Kazi, H., Igarashi, T., Zhao, S., & Davis, R. (2012). Vignette: interactive texture design and manipulation with freeform gestures for pen-and-ink illustration. ACM CHI. 1727-1736.

80. Kazi, R.H., Chua, K.C., Zhao, S., Davis, R., Lim, K. (2011). SandCanvas: A multi-touch art medium inspired by sand animation. ACM CHI. pp. 1283-1292.

81. Kennedy, J. M. & Silver, J. (1974). Perception 3, 313–322.

82. Kenyon H, (1998). Animating under the Camera, in Animation Magazine.

83. Khan Academy, http://www.khanacademy.org.

84. Kopf, J., Fu, C., Chohen-Or, D., Deussen, O., Lischinski, D. and Wong, T. (2007) Solid Texture Synthesis from 2D exemplars. In Proc. ACM SIGGRAPH 26, 3, 21- 29

85. Krueger MW, Gionfriddo T, and Hinrichsen K. (1985) VideoPlace—an artificial reality. *SIGCHI Bulletin*; 16(4): 35-40.

86. Lake, A., Marshall, C., Harris, M., & Blackstein, M. (2000). Stylized rendering techniques for scalable real-time 3D animation. ACM NPAR (pp. 13-20).

87. Lakin F. (1986). A performing medium for working group graphics. Proceedings of the CSCW, 255-266.

88. Landay, J. A. (1996). SILK: sketching interfaces like krazy. ACM CHI (pp. 398-399).

89. Landay, J. and Myers, B. (2001) Sketching Interfaces: Toward More Human Interface Design. IEEE Computer. 34(3):56-64.

90. Levin G, (2009). Audiovisual Software Art: A Partial History.

91. Levin G, (2000). Painterly Interfaces for Audiovisual Performance, in Media Laboratory, School of Architecture and Planning, Massachusetts Institute of Technology

92. Li X and J.M.Morshell. (1993). Modelling soil: Realtime dynamic models for oil slippage and manipulation. ACM SIGGRAPH.

93. Li Y, Hinckley, K., Guan, Z., & Landay, J. A. (2005). Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. ACM CHI, 461-470.

94. M. Tohidi, W. Buxton, R. Baecker, and A. Sellen. (2006). Getting the design right and the right design. ACM CHI, pp. 1243-1252.

95. M. Wattenberg. (2001). Sketching a graph to query a timeseries database. Ext. Abs. CHI, pp. 381-382.

96. M.D. Gross and E.Y. Do. (1996). Ambiguous Intentions: A paper-like interface for creative design, Proc. UIST, pp. 183-192.

97. Ma, C., Wei, L. and Tong, X. (2011). Discrete Element Textures. In Proc. ACM SIGGRAPH 30, 4, Article 62

98. Ma, C., Wei, Y., Lefebvre, S., and Tong, X. (2013) Dynamic Element Textures. ACM SIGGRAPH. Art. 90.

99. McCloud, S. (1993). Understanding Comics: The Invisible Art . Tundra Publishing Ltd.

100. McCloud, S. (2006). Making comics. Harper Paperbacks.

101. McCloud, S. (1994). Understanding Comics. Harper Paperbacks.

102. McCullough M, (1998) Abstracting Craft: The practiced Digital Hand. MIT Press.

103. Moscovich, T. (2001) Animation Sketching: An Approach to Accessible Animation. Brown University.

104. N. Gershon and W. Page. (2001). What storytelling can do for information visualization. CACM, vol. 44, no. 8, pp. 31-37.

105. N. Halper, M. Mellin, C. Herrmann, V. Linneweber, and T. Strothotte. (2003). Psychology and non-photorealistic rendering: The beginning of a beautiful relationship. Mensch & Computer 2003: Interaktion in Bewegung, pp. 277–286. Teubner Verlag, Stuttgart.

106. Narrain, R., Golas, A., Curtis, S. and Lin, M. (2009). Aggregate dynamics for dense crowd simulation. ACM Trans. Graph. 28(5):122.

107. Nelson Chu WB, Li-Yi Wei, and Naga Govindaraju. (2010). Detail-preserving paint modeling for 3D brushes. ACM NPAR.

108. NewTek Lightwave. www.lightwave3d.com.

109. Nice, C. (1997). Drawing in Pen and Ink. North Light Books.

110. Nienhaus, M., & Döllner, J. (2004). Sketchy drawings. In Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa (pp. 73-81). ACM.

111. Onoue K and Nishita T. (2005). An Interactive Deformation System for Granular Material. Computer Graphics Forum; 24(1): 51-60.

112. Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc..

113. Parent, R. (2012). Computer Animation – Algorithms and Techniques (3rd Ed) Chapter 5. Morgan Kaufmann.

114. Perceptive Pixel, http://www.perceptivepixel.com.

115. Popović, J., Seitz, S. and . Erdmann M. (2003). Motion Sketching for Control of Rigid-body Simulations. ACM Trans. Graph. 22(4):1034-1054.

116. Powering the Earth - CNN.com, http://edition.cnn.com/SPECIALS/2009/environment/energy.

117. PowerPoint Presentation and Slide Software - Office.com, http://office.microsoft.com/en-us/powerpoint.

118. Q. Michelle, Q. Baldonado, A. Woodruff, and A. Kuchinsky. (2000). Guidelines for using multiple views in information visualization, Proc. AVI, pp. 110-119.

119. R. Eccles, T. Kapler, R. Harper, and W. Wright. (2007). Stories in geotime. Proc. VAST, pp. 19-26.

120. Ramachandran, V. S. and Hirstein, W. (1999), "The Science of Art: A Neurological Theory of Aesthetic Experience", Journal of Consciousness Studies 6, 6/7: 15–51.

121. RedGiant.Trapcode Particular. www.redgiant.com/products/all/trapcode-particular/

122. Reeves, T. (1983). Particle systems—a technique for modeling a class of fuzzy objects. ACM Trans. Graph. 17(3):359-375.

123. Renaissance Art. http://en.wikipedia.org/wiki/Renaissance_art#Techniques

124. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., & Kafai, Y. (2009). Scratch: programming for all.Communications of the ACM, 52(11), 60-67.

125. Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. ACM Trans. Graph. 21(4):25-34.

126. Richard T, (1999). The encyclopedia of Animation Techniques.

127. RSA - Home, http://thersa.org.

128. Ryokai K, Marti S, and Ishii H. (2004). I/O brush: drawing with everyday objects as ink. In. ACM CHI, 303-310.

129. S. Bae, R. Balakrishnan, and K. Singh. (2009). ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. ACM UIST, pp. 151-160.

130. S. Bateman, R.L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks. (2010). Useful junk? The effects of visual embellishment on comprehension and memorability of Charts. ACM CHI, pp. 2573-2582.

131. S. Cheema, S. Gulwani, and J.J. LaViola. (2012). QuickDraw: improving

drawing experience for geometric diagrams. Proc. CHI, pp. 1037-1064.

132. S. Few. (2011). The Chartjunk Debate: A Close Examination of Recent Findings. Visual Business Intelligent Newspaper.

133. Salisbury, M., Anderson, S., Barzel, R. and Salesin, D. (1994). Interactive pen and ink rendering. In Proc. ACM SIGGRAPH, 101 - 108

134. Salisbury, M.P., Wong, M.T., Hughes, J.F. and Salesin, D.H. (1997). Orientable Textures for Image-Based Pen-and-Ink Illustration. ACM SIGGRAPH, 401- 406.

135. Sand Animation: Interview with Cako. http://www.sandanimation.com/page.php?2.

136. Santosa, S., Chevalier, F., Balakrishnan, R., and Singh, K. (2013). Direct space-time trajectory control for visual media editing. ACM CHI. 1149-1158.

137. Sato, M., Poupyrev, I., & Harrison, C. (2012). Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. ACM CHI (pp. 483-492).

138. Schmidt, R., Khan, A., Kurtenbach, G., & Singh, K. (2009). On expert performance in 3D curve-drawing tasks. ACM SBIM (pp. 133-140).

139. Schneider J, Kraus M, and Westermann Ru. (2009). GPU-Based Real Time Discrete Euclidean Distance Transforms with Precise Error Bounds. VISAPP.

140. Schödl, A., Szeliski, R., Salesin, D., and Essa, I. (2000). Video textures. ACM SIGGRAPH. 489-498.

141. Schwarz, M., Isenberg, T., Mason, K. and Carpendale, S. (2007) Modeling with Rendering Primitives: An Interactive Non-Photo Realistic Canvas. Non-Photorealistic Animation and Rendering, 15 - 22

142. Sewall, J. Wilkie, D., and Lin, M. (2011). Interactive hybrid simulation of large-scale traffic. ACM Trans. Graph. 30(6) Art. 135.

143. Seymour, P. (2003). The Artist's Handbook: A Complete Professional Guide to Materials and Techniques. il.

144. Shneiderman, B. (1997). Direct manipulation for comprehensible, predictable and controllable user interfaces. In Proceedings of the 2nd international conference on Intelligent user interfaces (pp. 33-39). ACM.

145. Simmons, G. (1992). The Technical Pen. Watson-Guptill.

146. SketchBook Pro. http://www.autodesk.com/products/sketchbook-pro.

147. Snibbe SS and Levin G. (2000) Interactive dynamic abstraction. In. ACM

NPAR, 21-29.

148. Song, H., Benko, H., Guimbretiere, F., Izadi, S., Cao, X., & Hinckley, K. (2011). Grips and gestures on a multi-touch pen. ACM CHI (pp. 1323-1332).

149. Stam, J. (1997). Stochastic dynamics: Simulating the effects of turbulence on flexible structures. Computer Graphics Forum. 16(3):C159-C164.

150. Summer RW, Brien JFO, and Hodgins JK. (1999). Animating Sand, Mud and Snow. Eurographics.

151. T. Strothotte and S. Schlechtweg. (2002). Non-photorealistic computer graphics: modeling, rendering, and animation. Morgan Kaufmann.

152. Tableau Public, http://tableausoftware.com/public.

153. Thomas, F., Johnston, O., & Frank. Thomas. (1995). The illusion of life: Disney animation (pp. 306-312). New York: Hyperion.

154. Thorne, M., Burke, D., and van de Panne, M. (2007). Motion Doodles: An Interface for Sketching Character Motion. ACM Trans. Graph. 23(3):424-431.

155. Tufte Edward, R. (1990). Envisioning Information. Cheshire (Conn.).

156. Turk, G. (2001) Texture synthesis on surfaces. In Proc. ACM SIGGRAPH, 347 - 354

157. Ura M, et al. (2009). A Paint Tool for Image Generation of Sand Animation Style. IEICE Tech.

158. Vandoren P, Van Laerhoven, T., Claesen, L., Taelman, J., Raymaekers, C., & Van Reeth, F. (2009) FluidPaint: an interactive digital painting system using real wet brushes. In. ACM Interactive Tabletops and Surfaces 2009, 53-56.

159. Vandoren P, Van Laerhoven, T., Claesen, L., Taelman, J., Raymaekers, C., & Van Reeth, F. (2008) IntuPaint: Bridging the Gap between Physical and Digital Painting. IEEE TABLETOP.

160. Victor, B. (2012). Inventing on Principles. CUSE.

161. W. Wojtkowski and W.G. Wojtkowski. (2002). Storytelling: its role in information visualization. European Systems Science Congress.

162. W.O. Chao, T. Munzner, and M. van de Panne. (2010). Poster: Rapid pencentric authoring of improvisational visualizations with NapkinVis. Posters Compendium InfoVis.

163. Wei, L. and Levoy, M. (2000). Fast texture synthesis using tree-structured vector quantization. In Proc. ACM SIGGRAPH, 479 - 488

164. Wei, L., Lefebvre, S., Kwatra, V. and Turk, G. (2009). State of the Art in

Example-Based Texture Synthesis. Eurographics.

165. Wilson AD, Izadi, S., Hilliges, O., Garcia-Mendoza, A., & Kirk, D (2008). Bringing Physics to the Surface. ACM UIST, 67-76.

166. Winkenbach, G. and Salesin, D. (1994). Computer Generated Pen and Ink illustration. *In Proc.* ACM SIGGRAPH, 91 - 100

167. Winkenbach, G. and Salesin, D.H. (1996). Rendering Parametric Surfaces in Pen and Ink. ACM SIGGRAPH, 469 – 476

168. Winnemöller, H. (2013). NPR in the Wild. In Image and Video-Based Artistic Stylisation (pp. 353-374). Springer London.

169. Winnemöller, H., & Bangay, S. (2002). Geometric approximations towards free specular comic shading. In Computer graphics forum (Vol. 21, No. 3, pp. 309-316). Blackwell Publishing, Inc.

170. Winnemöller, H., Olsen, S. C., & Gooch, B. (2006). Real-time video abstraction. In ACM Transactions On Graphics (TOG) (Vol. 25, No. 3, pp. 1221-1226). ACM.

171. Winnemoller, H., Orzan, A., Boissieux, L. and Thollot, J. (2009) Texture Design and Draping in 2D images. Computer Graphics Forum. In Proc. Eurographics 28, 4, 1091 - 1099

172. Wobbrock JO, Morris MR, and Wilson AD. (2009). User-defined Gestures for Surface Computing. ACM CHI, 1083-1092.

173. Wujec, T. (2011). Imagine, Design, Create: How Designers, Engineers And Architects Are Changing Our World.