ABSTRACT
This paper considers a new approach that allows for the creation of typographic works through gestural interaction. It describes a prototype application called TextDraw, which is designed to introduce new techniques for typesetting opened up by gesture and tangible manipulation. Previous work explored the implications of tangible-surface interaction in the context of Surface Games; specifically, how the combination of multi-touch and tangible objects can extend the paradigm of board games in new directions.[4] TextDraw also looks toward tangible-surface interaction and the idea that associating functions to gestures combined with new tangible tools will result in radically new methodologies for design, artistic and poetic expression of typographic works. Furthermore, while artworks created with TextDraw are presented, this paper focuses on the introduction / implication of our prototype.

Author Keywords  Typography, moveable type, gestural typesetting, tangible, multi-touch, interaction.

ACM Classification Keywords
H5.2.[Information Interfaces and Presentation]: User Interfaces --- Input Devices and Methodologies, Interaction Styles, Prototyping, Theory and Methods; J5.[Computer Applications] Arts and Humanities --- Fine Arts.

General terms  Design, Experimentation, Theory

INTRODUCTION
Our work is motivated by a vision that sees an incredible opportunity not only for experimentation, but also for a radical divergence from the methodologies of modern typesetting environments. Particularly, we see a limitation in modern editors for the layout of textual information, such as Illustrator and Word, which is the linearity of practice. We believe this limitation is a product of design for mouse and keyboard interaction based on the remediation of printing press techniques.

We situate our work by presenting a short background of experimental typography, leading up to the current state of the art. This is followed by a description of what we call gestural typesetting – using gestural interfaces for typographic design. We then introduce the functionality of TextDraw and show how it demonstrates multi-linear control of type attributes through gesture. Finally, we outline our ongoing research and present how we will extend our methodology into tangible-surface interfaces.

Figure 1 - Sketching a landscape with TextDraw

BACKGROUND / EXPERIMENTAL TYPOGRAPHY
Typographic experimentation in the deconstruction of formality provided a philosophical foundation for art movements of the 20s and 30s that resulted in a “sweeping away of exhausted conventions.”[11] Early examples that may be seen as the foundation for Dadaist principles, arose in the poetic works of Stéphane Mallarmé (concrete poetry) and Guillaume Apollinaire (calligrammes) where the visual form was as important as its wording (Figure 2).

In the 50s, a different sort of experimentation – focused on rigid structures, cleanliness, and readability – gave birth to the International Typographic Style, established by Emil Ruder, in Switzerland. Later, artist-typographers, like Wolfgang Weingart, took a mastery of Swiss typography and "blew it apart“[14] by creating works that were fundamentally sound in terms of formal rules but visually antagonistic in terms of the clean look usually attributed to the Swiss Style (Figure 3). The quick succession of these movements laid the foundation of both formal and experimental practice for the digital age of typography.

In the 80s and 90s graphic designers, such as April Greiman, and media artists, like John Maeda, began investigating new forms of typographic expression made possible through the computer. Since then, there has been an explosion of new work in the creation of generative
typefaces in which the glyph – the visual form of an individual letter – changes shape dynamically.\[1\]\[7\]\[12\] Other work has focused on the display of massive amounts of text resulting in a “rethinking” of the book. [10] Also, some research has explored kinetic typography and the possibility of giving character to type through programmable animation.[5][6]

![Concrete Poetry (left), Calligramme (right)](image)

**Figure 2 - Concrete Poetry (left), Calligramme (right)**

**GESTURAL TYPESETTING**

**Typesetting and the Model View Controller Paradigm**

The arrangement of type involves the selection of typefaces, point size, line length, leading (line spacing), tracking and kerning. The introduction of word-processing and digital typesetting revolutionized the practice of typography, but the metaphors behind the interfaces for such software are heavily rooted in those that applied to the printing press. Yet, despite the ease of which it is possible to change the look and style of a typographic piece, the process remains rigidly linear. For instance, if a designer wants to move a block of text and reduce its size, while maintaining the same text, the process might look like this:

*select drag tool → reposition text block → select size adjustment tool → readjust size of text block → select type tool → select text within the block → change font size*

This process involves a series of ordered actions that result in the changing of attributes associated with textual information. We believe this linearity to be a product of two things: these applications are, fundamentally, remediations of old technologies (e.g. the Gutenberg letterpress, typewriters) and are reliant on graphical user interfaces designed for mouse-based interaction.

The metaphors, and especially the interface of keyboard, mouse and model-view-controller make it difficult to dynamically change multiple attributes *at the same time*. New forms of interfaces open up the possibility to break through the linear paradigm of modern typesetting practice, and bring back an element of physicality that has been missing in the field for some time now. It is here where our research begins, currently looking at gesture with a focus on adapting the results to tangible-surface interaction.

**Tangible Interfaces for Typography**

Much work has been done in the development of graspable and tangible interfaces.[2][3][13] Significant aspects of these types of interfaces are: the ability to attach various functions to interface objects, as well as the affordance for more natural gestural object-oriented control.[2] This opens up the possibility of returning to a physicality of practice that was displaced by digitization. Also, there exists an incredible space for research because little work has been undertaken in the area of tangible interfaces for typography.

**Gestural Typesetting**

We are introducing the concept of applying *gestural interaction* to the field of typography through new forms of tangible interfaces. In general, gestural interfaces “are capable of recognizing meaningful configurations of strokes, including handwritten text, pointing and other stroke configurations.”[8] Rhyne highlights one specific advantage of this type of interaction is that “a single gesture can be equivalent to many keystrokes and mouse actions.” In more recent work, the use of gestural interfaces for navigation and manipulation of large information spaces has shown gestures mapped directly to specific functions allows for refined control over graphical space. [15]

Our work explores a combination of these two approaches, and uses pen-based interaction research to inform the development of tangible tools. Within the context of a pointer-based interface there remains a great deal of opportunity investigate the use of subtle tangible object orientation as input for non-linear typesetting.

For example, using a pressure sensitive pen interface allows for the variable adjustment of at least three metrics simultaneously. One possible combination of functions can translate into a drawing interface where the size of individual glyphs is controlled by *pressure*, placement is controlled by the *location*, and rotation is determined by *sweep/drawing direction*. (Figures 4 & 5)

The product of this research can be used to inform the design of new tangible objects. Examples that use the same kind of subtlety could be: an object whose presence, rotation and speed adjusts letter-spacing in a body of text. As well, the control of line-spacing objects may adjust shift text when they are flicked in different directions. In
particular, our research focuses on the ability to control the arrangement and attribution of text on multiple levels simultaneously, and the possibility of using gestural interaction together with tangible tools.

![Figure 4](image)

**Figure 4** – Sample mappings: a) pressure adjusts glyph size, b) x-tilt adjusts opacity, c) y-tilt adjusts kerning

In this section we describe our current prototype, TextDraw, and outline how it successfully illustrates some basic concepts of gestural typesetting. TextDraw is an application that allows for gestural layout of text by providing direct control over the manipulation and attribution of individual glyphs. To do this, the current prototype uses a pen interface that provides metrics such as speed, pressure, and x/y-tilt. This data is mapped to varieties of different typesetting functions, such as kerning, point-size, font weight, baseline height, and so on.

In TextDraw, the position and orientation of the pen adjusts multiple type attributes simultaneously, where in other programs (e.g. Adobe Illustrator) those same attributes must be adjusted through controllers in a linear fashion.

The current implementation has been developed as an artistic canvas, essentially a blank page, upon which a user can draw as if with pen, pencil or charcoal. For the moment, we are focused on uncovering new typesetting techniques, and less on fine adjustment and editing. However, we recognize this step as a logical one for future research. As a canvas, then, our initial goal has been to develop an application that allows for the creation of expressive works. Accordingly, the pen-based interface is natural and creates a low barrier to technology for new users.

When someone sits down with the application, and can almost immediately begin experimenting with gestural typesetting. The ease of use was noticeable as many artists from the Banff Centre were able to spend a significant amount of time (i.e. more than 30 minutes) working with the software. As we discuss further, in Informal User Observations (below), the flexibility of the software is able to accommodate a wide variety of artistic styles from painters and sketch artists to poets as well as graphic designers.

**Technical Description**

**Objective-C / Carbon**

The current prototype has been built using Cocoa, which combines Objective-C and Carbon. It also uses some modified open-source software for accessing pen data. The interface was designed using Interface Builder, and the application currently runs on OSX 10.5 and above.

**Canvas and Interface**

The canvas is a scalable window with a large drawing area and a minimal interface. Any text file can be chosen as the medium for drawing. As well, it is possible to load an image into the background and toggle its visibility for tracing. Figure 6 shows the standard color & font panels, and a double-slider object allows for setting the min/max point size, alongside some drawn lines.

**Recording Interaction**

The TextDraw application uses an event-based structure for recording interaction and updating the viewing area. On touch down of the pen, the system creates a new TDPath object and begins adding TDPPoint objects to the path, as the user draws. When the pen is removed, the system closes the path and performs glyph attribution and text manipulation to fill the drawn line with type.

**Data Structure: TDPPoint**

TDPPoint is a custom structure for recording necessary pen data. Each point contains a CGPoint structure and four float values representing x-tilt, y-tilt, pressure and barrel rotation. Additionally, a point is able to calculate the distance between itself and the location of another point.
**Data Structure: TDPath**

TDPath is a structure that accomplishes two things. First, it stores point data as an array of TDPoint objects. Second, it takes a specific piece of text represented as a mutable string, and adjusts and fits glyphs as close as possible to the line represented by its array of TDPoint objects.

**Figure 6 - Canvas, font & color panels with drawn lines**

**Glyph Attribution**

Associating typographic attributes to individual glyphs is a dynamic process because gestural manipulation changes the quality of a line on a point-by-point basis. There are two main considerations for calculating the appropriate attributes. First, the properties of an individual glyph are dependent on its distance from the beginning of the line in which it is included. Second, the distance a glyph will sit from the beginning of the line is directly dependent on the position and size of the previous glyph. So, properties need to be calculated and attributed one glyph at a time.

**Handling Finite Text**

Because a body of text is a finite composition – containing a certain number of words or characters – the attribution of text to TDPath objects poses three main issues:

1. It may happen that the end of the original text lies somewhere in the middle of the drawn line. In this case the original text is appended to the given text and attribution continues.
2. The length of a given text needs only to be as long as the area in which it is visible. If it is longer than the visible path, a substring representing the necessary characters is kept and the rest discarded.
3. To draw through an entire body of text, a reference to the last visible character of a path is stored so that the next path can continue at the appropriate location.

**Sweep Recognition**

The application has a basic recognition of drawing direction. When any part of a line is drawn left-to-right, text can be placed on the line and simply rotated to match the angle at a given point. If, however, any part of a given line is drawn right-to-left the application compensates by inverting the glyphs so that they read in the proper orientation. This recognition also applies to clockwise and counter-clockwise drawing direction.

**Background Images**

TextDraw allows for the selection of an image for tracing. An image is displayed in the background of the canvas, with reduced opacity – this makes it easier, say, if an artist wishes to create a text-based version of an existing picture or drawing. Figure 7 shows an original and a trace.

**Figure 7 – Scan of an original engraving by Gustave Doré (top), traced version using TextDraw (bottom)**

**INFORMAL USER OBSERVATIONS**

The open environment of the Banff Centre has provided ample opportunities to discuss and evaluate TextDraw with a wide variety of users including: artists, designers, authors, poets, media practitioners and critics. We have hosted several informal design sessions, with a variety of users, which informed some of the initial development of our application. In each session only the basic functionality of the prototype was introduced with little or no discussion about the purpose of our research – our intention was to allow new users to approach the software from a clear and uninformed perspective.

We briefly discuss two cases that had significant impact on the development of the initial prototype. Both artists presented herein are classically trained. As well, after the initial presentation of the software both artists were so inspired about the context and application of TextDraw that they each came back, voluntarily, over the course of some weeks to continue working with the software in their own creative styles. While the feedback from many other users...
was incredibly valuable for this work, a full discussion of their implications is beyond the scope of this paper.

The first artist spent several sessions documenting his use of the software while exploring various ideas for expression with text. At his request, the functionality of adding a background image for tracing was incorporated. His works with the software included using texts taken from current news events to create corresponding imagery – he mainly did this through tracings.

A second artist focused on the technique of gesture drawing – sketching figures and poses in short amounts of time. She requested the feeling of the pen incorporated more delicate control over multiple text attributes. In response to this, we refined memory allocation and response times of the interface, as well as incorporated dynamic control over kerning, point size and opacity in single strokes. The basic texts she chose were poetic pieces that she created in order to give more abstract feelings to the final artworks.

CONCLUSION
In this paper we have introduced a prototype application that illustrates an approach for the creation of typographic works that breaks away from modern practice. While pen-based interaction is not new, it is a very refined technology that is easily accessible for new users and still provides a rich platform for exploring gesture and expanding many of the ideas presented in this paper. As such, we are continuing to develop TextDraw in a direction that explores more concretely the effects of various gesture-function combinations for typesetting. We are also investigating the visual quality and materiality of text with regards to spontaneous optical composition and variations in typeface.

We have briefly labeled a new kind of interaction called gestural typesetting which builds from experimental research in the field of digital typography. As Simanowski points out, traditional forms of experimental typography have moved from print to digital media.[9] This is where we see the artistic products of TextDraw can be situated, as experimental pieces that expressively sweep away the convention and linear practice of modern typesetting.

FUTURE WORK
Through the development of new typesetting technologies, we will begin to introduce functionality that allows for the production of carefully designed typographic works. As well, focusing on gestural interaction will also support and inform the design of other applications for tangible-surfaces, such as Surface Games.[4] Future refinements to the current prototype will illustrate the versatility available in the combination of gesture and tangible interfaces. Furthermore, we see that extending the current prototype to a tabletop environment will bring new forms of physicality back to the practice of typography and lay a foundation for new exploration within the field of tangible design.

Our vision sees the current interface as a stepping-stone towards the production of tangible-surface interfaces. We are currently developing a prototype that extends TextDraw from pen to Surface Interaction and will focus on bringing two-handed tangible interaction to typesetting.

ACKNOWLEDGEMENTS
This research was funded by iCore Alberta and conducted in the Collaboration Lab at the Banff New Media Institute, Banff Centre. We acknowledge the many people involved with informal discussions, in particular Isabella Stefanescu and Tony Allard. We also acknowledge the Interface Culture department at the Linz University of Arts.

REFERENCES