Computational Aesthetics in Graphics, Visualization, and Imaging (2011)
D. Cunningham and T. Isenberg (Editors)

Artistic Canvases for Gestural and Non-linear Typography

Travis Kirton[1]

[1]Department of Interface Culture, University of Art & Industrial Design, Linz, Austria

Abstract

This paper presents an exploration of gestural and non-linear typography through the production of two software applications, TextDraw and TypeIs. Both were created through a media-art-research practice, wherein extended periods of development and artistic practice were exclusive of one another. This approach yielded applications which challenge contemporary typesetting methodologies, and produced new artistic works which exemplify gestural and non-linear typesetting techniques. This paper discusses the development of both software applications, the artworks made possible through their use, and situates the work within a history of experimental western typography in the 20th century.

Categories and Subject Descriptors (according to ACM CCS): J.5 [Computer Graphics]: Computer Applications—Arts, fine and performing

1. Introduction

Two canvases for the production of gestural and non-linear typographic works are presented. Both applications are computational tools programmed and used in artistic practice by the author himself. The first section of this paper outlines a very brief history of experimental typography in the 20th century, which serves to illustrate the people, ideas and aesthetics which been most influential. Section 2, presents the motivations behind this work highlighting technical research in the field of Graspable and Tangible interfaces which provide a basis for the development of both applications.

Sections 4-7 present the canvases, called *TextDraw* and *TypeIs* as well as artworks. The structure of presenting Canvas->Artworks->Canvas->Artworks reflects a media-art-resaerch practice where extended periods of development were followed by equal periods of artistic practice.

2. A Brief History of 20th c. Experimental Typography

The 20th century has seen an incredible amount of expansion in the field of Typography. Accompanying this expansion has been an equally abundant space for typographic experimentation the roots of which, influencing the work presented in this paper, date back to the turn of the century.

2.1. Deconstruction: Precursors of the Avant Garde

Lewis Carroll's *A Mouse's Tale*, originally published in his book Alice's Adventures in Wonderland, used the shape, form and position of words to strengthen the content, concept and construction of the poem. Decades later, in 1897, Stéphane Mallarmé published *Un Coup de Dés J'amais N'abolirais le Hasard*. This was a 20-page poem which used of innovative amounts of space and multiple typefaces. This publication is seen as the "single most striking precedent for avant-garde experiment with the visual form of poetic language."[Dru94] At the time of its publication, this piece went against all traditional rules of typographic design.

Shortly thereafter, Guillaume Apollinaire was expressing contemporary Avante-Garde theory through a series of visual poetic works he called *Calligrammes*. Similar to Mallarmé's approach, Apollinaire radically diverged from traditional typographic practice, and in the same style as Carroll, sought to create synthetic visual-textual experiences.

2.2. Returning to the Grid: Die Neue Typographie

Following the Avante-Garde, and based on the works of artists such as Schwitters and Moholy-Nagy, Jan Tschichold's book *Die Neue Typographie* championed the need for clean, functional designs and advocated the use of sans-serif fonts and asymmetrical structures. With typography having "es-

CAe 2011, Vancouver, British Columbia, Canada, August 5 – 7, 2011.

DOI: 10.2312/COMPAESTH/COMPAESTH11/063-070

caped from the book," Tschichold's manifesto highlighted a shift towards Modernist typographic style.[Spe83]

Leading this movement, through the 50s and 60s, were Emil Ruder and Armin Hofmann at the Basel school in Switzerland. Highlighting simplicity, exactitude and minimal necessity, this movement came to be known as the International Typographic Style. Eventually, Wolfgang Weingart would develop a practice that moved away from the traditional typographer's medium of lettersets towards an incorporation of new technology. Using photographic techniques, he explored a variety of distortional effects on letterforms which would ultimately lead him to develop what is known as New Wave typography. Weingart's work broke new ground in the field and inspired designers, like April Greiman, who would adopt his approach and bring it into the digital era of typography.

2.3. Contemporary Digital Experimental Typography

During the 80s, designers like April Greiman and Muriel Cooper were among the first to embrace computer technology in their practices. Greiman was producing software aesthetics which extended New Wave principles, and largely influenced the style of designers like David Carson who experimented with text as texture. Cooper was beginning to establish a vision for the future of a new medium. She described this medium as space whose "...outstanding characteristics are dynamic in real-time, interactive, incredibly malleable..."[Art05] This push was inspired by a belief in the non-linear, simultaneous characteristic of visual design. Though completely different in aesthetic and production, her approach was reminiscent of Apollinaire's theory of simultaneity. Cooper's vision eventually progressed under John Maeda, into the Aesthetics and Computation Group (ACG) which became a broadly focused group interested in the development of elements of reactive form. Maeda explored computation and interactive design with a large emphasis on typography.

In the late 90s, artists like Peter Cho and David Small were exploring typography in similar but very different aspects at the same time. Cho's work focused on generative typefaces and malleable glyphs, whereas Small's work explored the display of potentially infinite amount of content within the limited scope of books. Ben Fry's work has focused on the visualization of large sets of information, primarily genetic sequences. Joshua Nimoy's work, similar to Cho's, has illustrated the possibility of physical typefaces that change shape. Jason Lewis has developed a media-art-research practice through the creation of software for kinetic and generative typography.

3. Motivation

The history described above, brief as it is, outlines a series of artists, designers and technologists whose practices have influenced the work presented in this paper. Like them, we find comfort in an incredibly open space from which we can challenge paradigms of thought in long-established fields. We are also inspired by a vision that sees the development of new kinds of software as a medium for issuing those challenges. In particular, the practice presented in this paper suggests that through the desire to diverge from traditional methodologies, there exists a space where artistic experimentation can produce new technologies that can help us reexamine long-established paradigms.

Specifically, this work looks at a limitation in modern typesetting environments, such as Adobe Illustrator and Microsoft Word, which manifests itself as a linear practice.[KJO10] This research has come to see this limitation as a by-product of design for keyboard and mouse interaction. Furthermore, this work shares in the belief that such linearity occurs as many elements of digital typesetting environments have been developed as general remediations of printing press techniques.[LN09]

The technical approach behind this work finds its roots in two places: the use of computation as a medium for creativity, and the development of physical interfaces for high-fidelity interaction with software environments. The former can be seen in the work of contemporary computational artists like Maeda, Reas and Lewis, who develop their own programming languages, writing software through which they are able to express their unique visions. The latter can be seen in the development of Tangible and Graspable interfaces which provide more natural affordances for control and interaction than the keyboard and mouse. This kind of interaction has been applied to gaming, music, storytelling, and other computationally enhanced environment, successfully explored by many researchers.[Fit96, IB97, vHFA*07, JGAK07, Men06, WRM08] We share a vision which sees new kinds of software/hardware interfaces as opportunities for returning to the field of typography a physical quality which has gone missing since print moved to the screen.[LN10]

Finally, it is with great hope that this work highlights and reinforces an understanding for the unique combination of technical research supported by artistic practice, and vice versa. Though an artist may limited by the quality of the brush, the act of producing an artwork should move beyond the demonstration of technique.

4. *TextDraw*: A Prototype For Gestural Typesetting

Previous research has shown the development of *TextDraw* as a prototype for Gestural Typesetting.[KJO10] This application provides direct control over the manipulation and attribution of individual glyphs through the use of gesture. For instance, one might want to change the size, colour and position of a letter at the same time. To achieve this, it was necessary to develop an interactive system based on a data

structure which would easily record movement, position, angle, and pressure at the same time. We call this structure a *weighted baseline*.

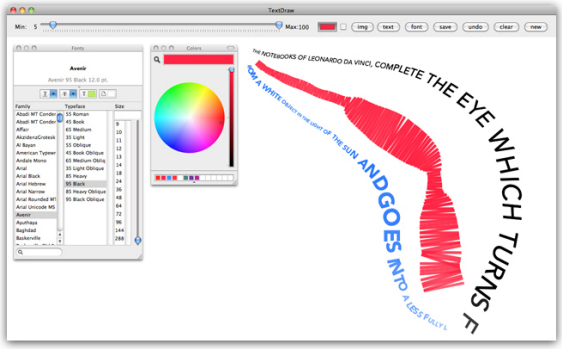

Figure 1: *The interface for TextDraw. On the left are panels for choosing font and colour. On the right two lines of text have been drawn in blue and black, with a third currently being drawn in red.*

The application is presented to the artist as a blank white canvas, a scalable window with a large drawing area and a minimal interface. There are options for changing typeface, colour, choosing text, importing a background image and exporting typographic works, as well a double-slider for adjusting the pressure sensitivity of the brush. As an artist is drawing, the canvas shows a stroke across the screen, varying in size depending on the pressure at a each point. When the artist lifts the pen, the system replaces the line with letters in the direction the line was drawn. Furthermore, the angle of the brush with respect to the canvas changes each letter's opacity.

The baseline records a variety of metrics on a point-by-point basis. When the artist removes the pen/brush from the canvas, letters are drawn sequentially, with their attributes determined by their position along the baseline. The position of each letter relies on the size, shape and characteristic of the previous letter. When the baseline is filled with letters, the application stops rendering and marks the final position in the text and then waits for the next drawn line to render.

Though texts are drawn through linearly, it is here where this technique becomes Gestural Typesetting as the position, rotation, size and opacity of a single character is are determined simultaneously by the stroke of an artist's brush.

5. Artworks Made with *TextDraw*

After completing the first implementation, approximately six months were spent working exclusively with *TextDraw*. A variety of studies were undertaken, first in the exploration of text as texture through comparisons primarily interested in the differences between lines drawn in ink, and those drawn with letters. Further comparisons in the visual qualities of a variety of typefaces were then conducted. As a series of first experiments, these studies focused on typographic translations of woodcut images.

The technique of woodcut was particularly interesting to mimic because an individual image could be laden with detail and depth, but the final aesthetic would come about only through the presence of negative space. Also, the effects of shading in a woodcut image are determined by the thickness and density of individual cuts. It would be possible, then to evaluate the visual effects of various typefaces, type densities, and font-sizes. More precisely, studying the gestalt effects of type as line was made easy when comparing against the woodcut because it became possible to see the difference between images made entirely of text versus those made entirely of solid forms.

5.1. Charon

Gustave Doré's image of Charon, from Alighieri's *Divine Comedy* was the main subject of these initial experiments. The image itself is extremely high in detail and very expressive, making it an interesting subject for translation between aesthetics. Furthermore, exploring the translation between solid lines an those made entirely of type was influenced by the work of artists like Schwitters and eventually Carson, who experimented with *text as texture*. The result of this study led to the feeling that clean, ultra heavy and dense fonts translate best as individual lines.

Figure 2: *A close-up of the caterpillar, smoking his pipe when he meets Alice.*

5.2. *Alice in Avenir & The Don's Imagination*

The next artworks engaged stories where the body of the image was made entirely of text from which it was written. The most important historical influences for these pieces come from Lewis's *A Mouse's Tale* and Apollinaire's *Calligrammes* where imagery, typography and content are merged into synaesthetic artworks.

Alice in Avenir was a series of three images, each a different chapter from Lewis Carroll's famous novel. Each image was drawn with the entire contents of its chapter, and visual quality of text was used more as a charcoal-like texture than converted lines as in the Charon works. A final study in the same technique was *The Don's Imagination*, another typographic translation of Doré's work. This piece used approximately 450,000 letters from the novel, and was drawn entirely by hand.

Figure 3: *A detailed view of Don Quixote's face from The Don's Imagination. Instead of tracing lines from Doré's woodcut image, shading with text as texture produced a more appealing aesthetic.*

5.3. Reflection on Artistic Practice with *TextDraw*

One of the main goals was to establish extended periods of research and development in rhythm with equal periods of artistic practice. Having initially developed *TextDraw* over the course of some months, the artworks presented were created during an equal period of focused artistic practice. This extended time spent devoted to creating new works made it possible to develop a deeper understanding of the software and the technique of Gestural Typesetting.

In the end, having worked in depth with the software provided a solid base of issues which a new phase of development could address. Though this process may appear to be similar in nature to iterative or agile development processes, we believe that the extended period away from development with focused artistic use of the software provides a richer experience for considering aspects of the project. In some cases, stepping away from the practical side for a long time even allowed for a more philosophical and considerate approach to the actual nature of the project.

6. *TypeIs*: A Prototype for Non-linear Typography

Having uncovered major limitations inherent in *TextDraw*, it was clear that a rebuilding of the underlying software was necessary. A new application was developed, extending previous techniques, with improvements that manifested as: a sophisticated text management system, document zooming and navigation, and character-by-character rendering in real-time. Through the previous period of artistic practice it also became clear that our approach was really questioning the methodologies and the nature of digital typesetting. With respect to the principle investigation of this work, the new application was named *TypeIs*.

6.1. Text Storage

The most significant development for handling text was a sophisticated text storage system. The major reason for this requirement was that rendering was directly affected by the length of the text file currently being drawn. The longer the file, the longer the rendering time and a greater dissociation between the act of drawing and its visible result. Furthermore, *TextDraw* had a strictly linear access to each character in a file. Developing a new text management system would not only be more efficient, but would also provide non-linear access to retrieving text from loaded files.

The text storage system in *TypeIs* offers many advantages. The first of which is that, from text and word documents to binary data files, it allows any kind of file to be loaded. This creates an ample flexibility for typesetting with any source whatsoever. From an artistic perspective this is important because it removes limitations of working purely with language-based sources. An artist could potentially draw with a textual interpretation of an image over top of itself, or create a typographic piece consisting of an entire movie.

When a file is loaded, it is broken up into manageable parts. Data files are converted to textual formats and then broken up into individual strings; text files are broken up on a paragraph-by-paragraph basis. The system is able store entire texts and work quickly with fractions of them. The outcome of this improvement correlates directly to the visual feedback of the system; for example, the application doesn't have to read through an entire book every time a line is drawn.

Another advantage of this system is that sections of texts can be accessed independently such that the last section of a text can be called as easily as the first, or as any other in between. To do so, the system provides methods that access the current section, the next section, or any specific section if they are available for drawing and so introduces Non-Linearity to the act of typesetting.

6.2. Line Management

TypeIs employs a line management system which bridges interaction and information, and is the mechanism for accessing individual characters from a body of text. Its major function is to receive user input and generate weighted baselines.

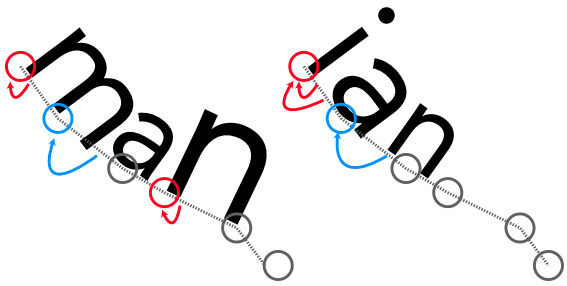

Figure 4: *An example of 2 words placed on the same baseline. Notice the letters a and n are different sizes between the word man and the name ian. This is because the letters m and i have different widths.*

While doing so, it accesses individual characters from a body of text, calculates and sets each character's attributes based on the line that is currently being drawn, then sets those characters to be rendered. It does all of this in real-time.

The line manager provides logical access to text on an as-needed basis. When an artist starts drawing a new line the application needs to access the nextAvailableParagraph with which it can begin drawing. Similarly, for any line that is currently being drawn the manager provides access to the nextAvailableCharacter in the paragraph bound to that line. As each character is drawn, the next one is called until the last one is drawn and the paragraph is marked complete. It can be that when one section is complete the next logical step would be to draw characters from the following paragraph. However, an alternative method could be to choose a random point in a body of text.

In code, the process for real-time line management and drawing looks like this:

```
if(mousePressed) {
  create a new line
  get the nextAvailableParagraph
  mark it as beingUsed
}

if(mouseDragged){
  create a new point
    add the point to the line
  get the nextAvailableCharacter
    set character attributes based
    on its position along the baseline

  if (no more characters in text) {
    mark the paragraph complete
    get the nextAvailableParagraph
  }
  else if (character fits on the line)
    draw the character
  }
  else {
  wait until line is long enough
    to draw the character
  }
}

if(mouseReleased) {
  if (text has characters) {
    mark paragraph as available
  }
  else {
    mark the paragraph complete
  }

}
```

6.3. Data Structures

There are four main data structures that provide the basis for *TypeIs*, two of which have evolved from *TextDraw*; those being TIPoint and TIBaseline. Through the development of a real-time system it was essential to create the TIString structure for formatting text and providing access to individual characters. The TICharacter structure was designed to provide a means for saving attributes and passing the letter between the text manager and the rendering system.

6.4. Data Structure: TIPoint

The TIPoint is similar to the TDPoint in that it is a structure that provides the basis for generating a weighted baseline. In anticipation of creating a playback function and in addition to origin, distance and angle, this new structure also takes a timestamp relative to the moment the first mark an artist makes on the canvas. Improvements over the original TDPoint include a variety of initialization methods, including those for confirming the proper setting of each variable in a given point.

6.5. Data Structure: TIBaseline

The most important structure in *TypeIs*, is the representation of a weighted baseline herein called TIBaseline. The reason for its importance is not so much that it is a series of weighted points, but that it is structured in such a way that it provides the mechanics for Gestural Typesetting. It not only provides the same imaginary line on top of which letters in a traditional typographic composition sit, but is also the basic reference for determining the visual attributes of individual characters.

The TIBasline functions such that the qualities of an individual letter are determined by its relative distance along the line.The structure contains a variety of methods with the

Figure 5: *A section of Panoramically Coloured, one of the works from TypeCity:Istanbul. This section contains approximately 125,000 of a total 250,000 characters from the original print.*

general title: attributeAtDistance. In order to properly attribute a character, its distance from beginning of the line is calculated based on the position and size of the previous letter.

The baseline always maintains a reference to a section of text by being dynamically bound to a TIString object from which it can request characters. As an artist draws, points are recorded and the baseline determines how much visible space is available. As the artist continues drawing, and as every new point is added, the baseline is able to determine if the next character will fit along the drawn line.

6.6. Data Structure: TICharacter

The TICharacter is a basic structure that facilitates passing attributed characters to the rendering context of the canvas. The object itself contains values for location and rotation angle, a copy of the character to render, and a set of visible attributes. After it is created with an individual character and specific attributes, it is composited into the rendering context, which could be for on-screen viewing or otherwise off-screen to a PDF or image file.

6.7. Data Structure:TIString

The fourth fundamental structure is used to process raw text and store it on a string-by-string basis. In general, if a typically formatted body of text is used then it will be broken into individual paragraphs, and a TIString object will be created in turn for each paragraph. Maintaining references to total character count, and caret position, this object tracks the used and remaining portions of a piece of text; furthermore, there are methods for accessing both the current and next characters.

An essential characteristic of this structure are the use of two flags. The first marks whether or not all the characters in a string have been requested, the second marks whether or not a string is actively bound to a baseline. The combination of these two flags lends to the determination of 3 particular states for the string object. These states are:

Available: A string is available if it is not complete and it is not being used.

Being Used: A string is being used if it is currently bound to a baseline. If the string has been returned by the line manager's nextAvailableParagraph method it is being used. This state can change back and forth until all characters have been requested. If a string is being used and its baseline is finished being drawn, the baseline sends the string a message. If there unused characters in the string it is again marked available. The next time it is bound, the new baseline will begin calling letters from the string's current caret position. If there are no remaining characters it sets itself complete.

Complete: If all characters in a string have been requested, it is marked complete and made unavailable and unusable.

6.8. Reflection on the development of *TypeIs*

The importance of the four data structures that have just been presented is in their combined ability to provide non-linear, gestural control in typesetting. This aspect can be applied to thinking artistically and experimentally about new kinds of aesthetics for typography. Even though the original works created with TextDraw are reminiscent of Lewis and Apollinaire's works, the computational nature of the canvas already influences the aesthetics of these works towards those beyond the pen and pencil. Furthermore, the stylus / brush interfaces TextDraw and TypeIs are not at all limitations of the software. The TIBaseline can be created with any kind of interface which allows for multimodal control. In terms of creative expression it is possible to see a movement towards blocks and bodies where tangible and performative interfaces can create new kinds of typographic aesthetics.

7. Artworks Made with *TypeIs*

The first pieces created with *TypeIs* were, like those presented above, experimental print works. However, the canvas itself became sophisticated enough that it was possible to create products which were embodied in a variety of different mediums including: interactive installations, sculpture, and performance.

7.1. *This Is Not Our Paradise*

The first exhibition of *TypeIs*, called *This is Not Our Paradise*, was an installation that engaged visitors by letting them draw text onto images. As a visitor walked up to the screen, an image was presented to them onto which they could leave their mark. Drawing on top of the image, letters and words were left behind like ink on paper. The content of the words was ever changing and could be explored by continuously drawing. Just as an image or an object might last longer than the words people express about them, the letters that people drew onto the installation began to fade after a while, eventually vanishing. This installation extended the ideas behind *TypeIs* by seeking to explore ephemeral relationships between image and text, between districts and dialogue, between the things people see and what they say.

7.2. Studies in Colour

The first print-based studies using *TypeIs* shifted from black and white to colour. Also, considering the new technical improvements these pieces attempted to push the limits of the application. Rather than splitting up images to be recombined using another software, two new pieces were created using long texts and oversized images. In both cases the results were positive and the new implementation successfully managed tens of thousands of letters instead of the mere hundreds that *TextDraw* was able to handle at a single time.

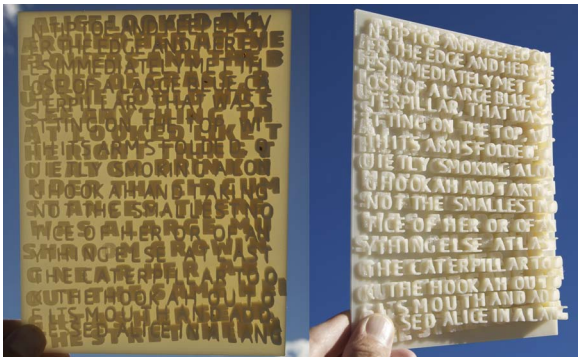

Figure 7: *A printed type sculpture, with two sets of text overlaid in different fonts.*

7.3. Physical Type

A third study moved text off-screen, off the printed page and into a sculptural form. The piece was typeset using *TypeIs* and transformed into a physical sculpture via 3D printing. Using two typefaces, the work layered two sets of text with the thin face extending out of the words set using ultra heavy font. This piece explored the aesthetics around the translucency of material typography.

7.4. *TypeCity:Istanbul*

The most recent work, created by the author, was a print-based/interactive typographic representation of Istanbul. *TypeCity:Istanbul* uses writings and data published by the London School of Economics (LSE), by drawing them into photographs of the city itself. Nearly all publications concerning the city of Istanbul were written by academics, architects and thinkers from around Turkey. The installation was a combination of three print works presented side-by-side with an interactive performance version of *TypeIs*. The print component was made of up original large-format works, each consisting of a different literary theme taken from the publication *Istanbul: City of Intersections*.

7.5. *Trace*

Recently both *TextDraw* and *TypeIs* were used to generate the aesthetics for a contemporary ballet called Trace. In this performance, dancers moved fluidly throughout space as text appeared and vanished in their wake. Joan Karlen choreographed *Trace*, the first example of another artist using both canvases for their own expressive vision.

8. Reflection on Artistic Practice with *TypeIs*

The first period of artistic exploration and practice was an opportunity to really dig into *TextDraw* and which provided a solid base of experience for achieving new technological developments. These developments greatly improved the experience of working with the new system as well as the quality of works produced with the application. One of the most important results to arise through this period of artistic practice was an understanding of the potential to introduce a performative aspect for typesetting. When a dancer can create a typographic work as easily as a typographer, the potential for new forms of typographic expression becomes enormous.

9. Conclusion and Future Work

Two canvases, *TextDraw* and *TypeIs*, for gestural and non-linear typesetting have been shown. Their effectiveness use for the production of new kinds of creative expression can be seen in the artworks which have been presented, which

Figure 6: *A dancer on stage. Typographic visuals, from the contemporary ballet Trace, were created with TypeIs.*

exist across a variety of media. Furthermore, a media-art-research practice has been presented where focused and separate periods of artistic production and research deeply inform one another. Furthermore, work will continue to investigate the nature of digital typesetting softwares and move towards establishing new approaches which can only be explored through computation. Finally, this work sees interactive environments where tangible objects, and multimodal interaction playing vital roles in the future of typography.

10. Acknowledgements

The author would like to acknowledge the support of the Department of Interface Culture at the University of Art and Industrial Design in Linz, Austria.

11. Bibliography

[Dru94] Drucker, J., The visible word : experimental typography and modern art, 1909-1923. 1994, Chicago: University of Chicago Press.

[Spe83] Spencer, H., Pioneers of modern typography. 1983, Cambridge, Mass.: MIT Press.

[Rai03] Raizman, D.S., History of modern design : graphics and products since the Industrial Revolution. 2003, London: Laurence King.

[Art05] Art Director's Club, Art directors annual. 2005: RotoVision.

[Kin04] Kinross, R., Modern typography : an essay in critical history. 2004, London: Hyphen Press.

[Wei00] Weingart, W., Typography : My way to typography. 2000, Baden/Switzerland: Müller.

[KJO10] Kirton, T., Jennings, P.L., Ogawa, H., TextDraw: a Prototype for Gestural Typesetting, in Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction. 2010, ACM: Cambridge, Massachusetts, USA. p. 193-198.

[LN09] Lewis, J. and B. Nadeau, Writing with Complex Type. 2009.

[Fit96] Fitzmaurice, G.W., Graspable User Interfaces, in Department of Computer Science. 1996, University of Toronto: Toronto. p. 167.

[IB97] Ishii, H. and B. Ullmer, Tangible bits: towards seamless interfaces between people, bits and atoms, in Proceedings of the SIGCHI conference on Human factors in computing systems. 1997, ACM: Atlanta, Georgia, United States. p. 234-241.

[vHFA*07] Hoven, E. van den, Frens, J., Aliakseyeu, D., Martens, J.B., Overbeeke, K., Peters, P., Design research & tangible interaction, in Proceedings of the 1st international conference on Tangible and embedded interaction. 2007, ACM: Baton Rouge, Louisiana. p. 109-115.

[JGAK07] Jorda, S., Geiger, G., Alonso, M., Kaltenbrunner, M., The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces, in Proceedings of the 1st international conference on Tangible and embedded interaction. 2007, ACM: Baton Rouge, Louisiana.

[Men06] Mendels, P., Interactive Story Structures for StoryToy. 2006, Koninlijke Philips Electronics: Eindhoven.

[WRM08] Wu, C.-S.,Robinson, S.J., Mazalek, A., Turning a page on the digital annotation of physical books, in Proceedings of the 2nd international conference on Tangible and embedded interaction. 2008, ACM: Bonn, Germany.

[LN10] Lewis, J.E. and B. Nadeau, Post PostScript please. Digital Creativity, 2010. 21(1): p. 18 - 29.