Kairos
A Journal of Rhetoric, Technology, and Pedagogy

From Street to Software: How a Lettered *Flâneur* Invented a Hybrid Rhetoric

David M. Rieder, North Carolina State University

Abstract

This essay contains four sections. In the first, "Inventing a Writing Process," I narrate my inventional process, which culminated in the alphabetically arranged, computationally generated image-texts. In the second section, titled "Hybridizer.as," I offer a brief summary of the software program that I wrote with ActionScript 3 (AS3) as well as a list of the text strings that were used to generate each individual image-text. In the third section, "Hybridity, Cultural and Computational," I associate the image-texts with the definition and concept of hybridity. Since hybridity is computer-generated by the software program, this section includes an explanation of some of the key programming concepts and strategies in Hybridizer.as. In the fourth and last section, I emphasize the role of chance in computational new media related to composition.

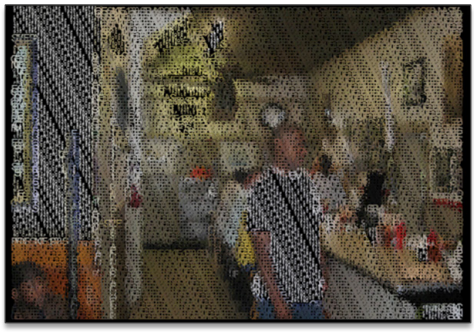

Image-Text for the letter Q, Clyde Cooper's BBQ, Raleigh, NC

Throughout this essay, the roles of the visual and computer arts are presented as an interdisciplinary source for new approaches to writing. As compositionists delve more deeply into the material and technical dimensions of digital media, the contemporary arts should be valued as a source for new approaches to hybrid forms of writing and textuality.

Rieder, David M. (2010). From street to software: How a lettered *flâneur* invented a hybrid rhetoric. *Kairos: A Journal of Rhetoric, Technology, and Pedagogy, 14*(2). http://kairos.technorhetoric.net/14.2/topoi/rieder

1. Inventing a Writing Process

This project was inspired initially by French artist Franck Scurti's work titled *Topo-Typographie* (2003) in which the letters of the alphabet are employed as a visual technology for exploring the city of Paris. Published in the form of a foldout map, Scurti invited his users to explore the city's streets by walking one or more of the 26 areas on the map on which the artist has traced a letter and for which he'd developed a short narrative of directions. For example, for the letter G (Fig. 1), Scurti instructed us to take the subway to Jourdain in the 20th arrondissement, and then to walk up and down five streets in that neighborhood in order to "write" the letter with our steps. Beginning on the Place du Guignier, the G is written by walking counter-clockwise through a neighborhood defined by three streets, the Rue de l'Ermitage, the Villa de l'Ermitage, and the Cité Leroy. The straight and curving lines that Scurti drew over a map of the city define our movements, and the sights and sounds associated with those movements become a part of the meaning of that letterform.

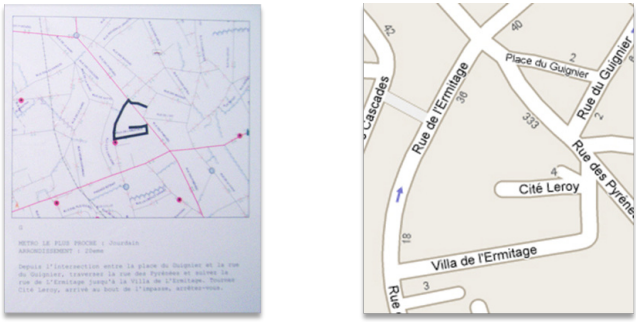

Figure 1. On the left, the letter G from Franck Scurti's Topo-Typographie*; on the right, a screenshot of the streets in Google Maps*

Scurti's project fascinated me for several reasons. First, the dual purpose of the lines, which represent simultaneously alphabetic and cartographic meaning, foregrounds the visual/material aspects of writing as generators of meaning. Like the awkwardness of a child's handwriting, which draws attention to writing's extra-verbal character, the skewed and oblong letters in Scurti's project emphasize the fact that writing, like cartography, is one of the many arts of the line. As a line art, the shapes of the alphabetic characters in Scurti's project are valued both over and above their traditional, *logocentric* uses—and beyond the aesthetic realm. They are used to discover experiences in and about place.

Second, *Topo-Typographie* adds a provocative twist to the form of urban exploration known as *flânerie*. In her book titled *Geographies of Writing*, Nedra Reynolds (2006) introduced the *flâneur* as an urban figure who "embodies the spatial practices of walking as writing, writing as walking" (p. 70). She explained that *flânerie* (the *flâneur*'s method of observation) is visually attuned to the complexities of urban life. As a person strolls through urban spaces, he observes, collects, and catalogs the experiences in which he finds himself immersed.

The observational method that the *flâneur* embodies is valued as a way to document the complex relations of life epitomized by the crowd. Like a mirror, the

flâneur's engagement with urban space is reflective. If, as Reynolds contended, the *flâneur* embodies a spatial practice in which walking is *like* writing, I contend that the observational walks that the *flâneur* takes are expository. In other words, the *flâneur* follows the organic patterns of urban composition: He creates nothing new, but rather reflects the complexity of what is already there.

Scurti's project, in turn, parallels the *flâneur*'s perambulatory engagement with urban space, but the "walked" letters complicate the expository approach. Scurti's "lettered" *flâneur* does not merely reflect what he sees, as the letters introduce vectors of purpose, direction, and meaning. Rather, the letters literally shape the reality that is experienced and observed—the shapes of the letters intervene to change the ways in which the city's flow is experienced. In this way, Scurti's project turns *flânerie* into an active form, one that can be related to narrative.

Finally, Scurti's project provides a novel response to Geoffrey Sirc's call for a "new urbanism" in composition studies. In Sirc's (2002) essay titled "Writing Classroom as A & P Parking Lot," he echoed the alternative relationship to writing described above when he wrote, "I prefer writing as road map to strange, new places over writing that simply charts again the same, well-worn ground" (p. 197). In that essay, Sirc challenged compositionists to break free of the introverted, ivory-mannered values of academic discourse in order to include the everyday logic of the street and pop culture in our pedagogies. While he does not specifically advocate extra-curricular writing projects such as Scurti's, Sirc's repeated calls for a renewed focus on the street as a place of discovery and exploration easily translates to Scurti's use of the alphabet.

For the above-mentioned reasons, I purchased one of Scurti's maps. When I returned to Raleigh, NC, I decided that I would develop an alphabet of my city. My initial hope was to use my own alphabetic explorations as field notes for an argument about place-based writing. Following Scurti's method, I traced the 26 characters of the alphabet on a street map of the city. Like Scurti's project, my choices in the spaces for letters were driven by shape over content. The content or text that resulted was derived from my direct experience writing that space. Armed with a camera, pen, and pad of paper, I visited each space and collected data during each experience. I took notes of my impressions and of details from the neighborhoods, like the names of stores, ads in store windows, graffiti, and even conversations that I overheard. I took photographs of people and places that seemed to epitomize that moment in each location. The walks were an inductive or experimental research method based on the alphabet as an exploratory heuristic.

My alphabetic explorations of Raleigh led to a realization about the visual/material dimensions of writing. The cartographic lines on the city map divided the alphabet into two typographic groups. The shapes of the lines on the map forced the shapes of the letters in two types of neighborhoods. One was suburban, and the other was urban. Letters with curving lines, such as Cs, Ds, Gs, and Ps, traced more easily over the curving streets and cul-de-sacs of Raleigh's suburban streets. Letters comprised of parallel and perpendicular lines, such as Es, Fs, Hs, and Ts, reflected the configurations of streets on the downtown street grid. This realization led to a new way of reading my city—or the city that I experienced—and I began to recognize lettered shapes throughout the urban and suburban areas that I tra-

versed in Raleigh. Lettered shapes appeared in all kinds of objects and architectural details: The city was transformed into a *texturology*.

In his book titled *Alphabet City*, artist Stephen T. Johnson (1995) used the term *urban compositions* to describe the emergence of lettered shapes from among the complex angles, shadows, and lines in an urban space. In the following excerpt, Johnson explained how his interest in urban compositions led to the paintings reprinted in his book (see Fig. 2). Johnson wrote,

> The idea for Alphabet City came to me while I was walking along the city street. I noticed an ornamental keystone that looked like the letter S. Then suddenly I saw the letter A in a construction sawhorse and the letter Z in fire escapes. At that moment, it became clear that in urban compositions I could discover the elements that form the letters of the alphabet. (p. 1)

Figure 2. The letter A in Stephen T. Johnson's book of paintings titled *Alphabet City*

Johnson's discovery of a means of reading urban scripts echoes my own. The more attention I paid to the visual/material character of the alphabet, the more I noticed the elements of literacy present in structural forms. Moreover, extending Michel de Certeau's (1985) claim in "Practices of Space" that the twists and turns of pedestrian movements can be equated with the turns of words and phrases in rhetoric, I experienced a city in which the alphabet emerged from the juxtaposition of activities and the wide range of forms in which those activities are materialized. This materialist turn in my experience of both writing and walking the city led to a repurposing of my method. Instead of using an approach such as Scurti's, which would culminate in a traditional academic text, I decided to develop a form of textuality representative of my experience: a hybrid, visual-verbal form of argumentation.

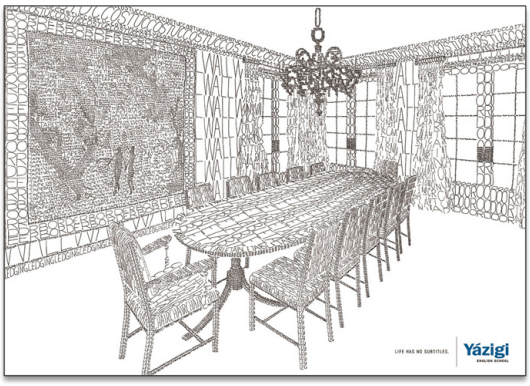

Figure 3. Print advertisement for Yázigi English School

My search through various hybrid forms of textuality led me to *micrography* or *microcalligraphy*, which is a hybrid visual-verbal form of text art in which letters are used to develop pictorial shapes. While it is attributed originally to tenth-century Jewish scribes who would combine images with mi-

niscule words from scripture, it is still used today. Figure 3 is a print advertisement for the Brazilian Yázigi English School constituted microcalligraphically.

Brendan Dawes pursued a similar typographic experiment in the computational realm through an open-source programming language called *Processing*. Processing is a language that was recently developed as a computational medium for visual artists and computer scientists who want to explore the possibilities of computers and art. In "Holy Typography, Batman!", Dawes (2006) reconstructed some of the image sequences from the 1960s television series *Batman* using letters and words. In Figure 4, letters play the role usually reserved for pixels in raster-based digital images. Dawes' method of using alphabetic characters to rewrite the original images provided me with a hybrid way of expressing my own experience of Raleigh's lettered spaces. Rather than use Processing, however, I developed a version of Dawes' project in Adobe Flash CS3 using ActionScript 3 (AS3), which I describe in detail in the next two sections of this essay.

Figure 4. Screenshot from Brendan Dawes' "Holy Typography, Batman!"

1.1 Final note

As I developed the software program that generates the hybrid, micrographic images of Raleigh, I returned to the street. In the place of Scurti's method, I adopted one described by the performance art group Wrights and Sites (Hodge, Smith, Persighetti, Turner, & Weaver, 2003). The group describes itself as "artist-researchers committed to producing experimental, site-specific work across a range of media." Among the works they've developed are a series of "anti-tourist" handbooks which they call *Mis-Guides* (see Fig. 5). In the following description of this genre of anti-handbook, Wrights and Sites underscored the importance of the personal experience as well as of the active role of the walker as a maker of meaning:

> A *Mis-Guide* often takes the form of a guide book or a map. It suggests a series of walks and points of observation and contemplation within a particular town, city or landscape. Unlike an ordinary guide book, it is guided by the practice of mytho-geography, which places the fictional, fanciful, fragile and personal on equal terms with 'factual', municipal history. Au-

> thor and walker become partners in ascribing significance to place. (p. 2)

In the *Mis-Guide* that they developed for the city of Exeter, England, Wrights and Sites described an activity titled "A to Z of Your Street." Wrights and Sites introduced the activity as follows:

> You may have taken many photographs of events inside the place where you live but do you have any shots of the street where you live? Make a photo A to Z of the street starting at your own front door. For example, W = Wall, X = Crossroad, Y = Yellow Lines. (p. 12)

This method interested me because it seemed more in line with the goal of associating one photograph with each letter. I returned to the streets to find 26 photographs of people, places, and events representative of the 26 letters of the alphabet. Since I knew that I would need a string of words to generate micrographic texts from each photograph, I recorded any words or phrases paradigmatic of the scenes in which the photographs were taken. As I decided which photographs I would use for each letter, I rewrote some of the code in the software program I'd written in AS3 in order to generate a compelling visual-verbal form of micrographic text.

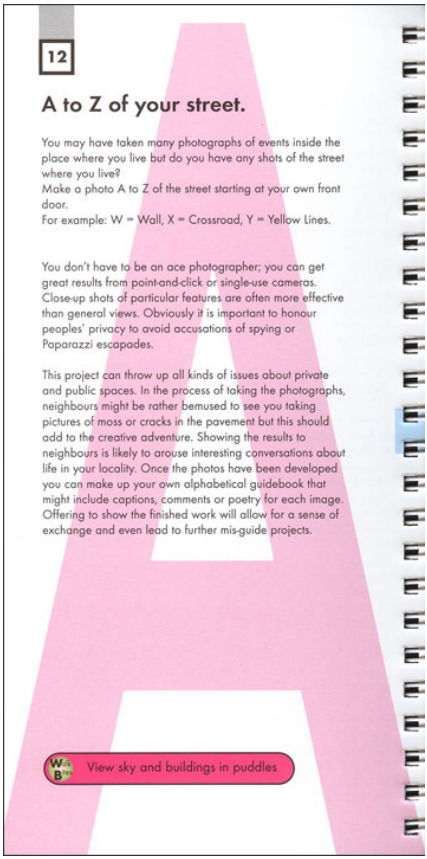

Figure 5. From their *Mis-Guide,* Wrights and Sites' hybrid use of the alphabet is a method that is supposed to lead to deeper engagements with place.

Throughout this description of my invention process, typography has been an essential part of my extra-verbal explorations of both place and digital textuality. Both Scurti's and Dawes' creative use of type inspired me to explore the possibilities of writing both after the page and after the phonocentric voice. During my process, the materiality of writing was made visible**,** which simultaneously made it a generative medium for new modes of communication and creative expression.

Based on my experiences developing this project, I concur with John Trimbur's call for compositionists to both study and teach the typographic arts—although my reasons are somewhat different. In Trimbur's (2002) article titled "Delivering the Message," he argued that the visible or material dimensions of writing were largely neglected during the process movement. Post-process theorists of writing have endeavored to *rematerialize* writing, and Trimbur argued that the typographic arts are a contribution to that effort. One of the reasons is described in the following quote:

> Typography enables us to *see* writing in material terms as letter forms, printed pages, posters, computer screens. It helps to name the available tools of representation that composers draw on to make their means of production. (p. 192)

While I agree with Trimbur's explanation, I would add the following observation regarding typography based on my experiences developing this project: The ability to see writing in material terms provides the basis for new methods of invention. Both Scurti and Dawes demonstrate how the material aspects of written verbal discourse can serve as the basis for new forms of invention both in and outside new media. During my process, their approaches helped me develop a materialist perspective that opened up new ways of valuing writing.

2. Hybridizer.as

Hybridizer.as was written in AS3, which is a high-level language popularly associated with the Adobe Flash and Flex applications. The program uses a combination of visual and verbal data to generate the hybrid image-texts. The verbal data was derived from one of two sources. For the letters B, C, I, K, N, P, Q, S, and Y, the text was part of the scene in which the photograph was taken. For example, the capitalized phrase, "IT PAYS TO BE WELL GROOMED" for letter N, was written on the door to Nicholson's Barbershop. I overheard the lyrics, "The Lord knows my name/I am not forgotten," as I took the photograph for the letter S. And for letter P, the phrase "Deliciosas Pupusas y Tacos" was written on the side of the truck from which local residents bought their lunch one Saturday afternoon during a soccer match. For the rest of the letters, I chose a string of words that had thematic relevance. For the letter A, which was based on a photograph of an Alpha Kappa Alpha sorority sister, I chose the organization's creed. For the letter G, which is based on a photograph of the graves of soldiers who fell at Gettysburg, I chose Lincoln's Address (1863/2005). And for the letter R (see Fig. 6), which is based on a photograph of the railroad tracks leading into the downtown Amtrak station, I chose a few lines from James Brown's (2004) "Night Train."

Figure 6. Image-text for the letter R based on lyrics from James Brown's "Night Train."

Almost all of the photographs in the project were taken during the summer and fall of 2008. With the exception of the photograph of "Bridezilla" at the Kirby Derby (see the letter Z), I took the pictures with an inexpensive digital camera, cropping and optimizing them to a size of 300 x 200 pixel JPG images without compression and at maximum quality. The image-texts in this project are JPG images made from screen captures of the software program's output. The reason I used screen captures is that it takes 30-60 seconds for the Flash Player (in a browser) to generate an image-text, which is too long a wait.

Hybridizer.as is included in the Downloads section of this project. It is offered as free software under the GNU General Public License. The .as file is accompanied by an .fla file, which you can use to execute the .as file, and a 200 x 133 pixel version of the JPG image that was used to generate the image-text for the letter Z. If Hybridizer.as is assigned to the Document class in the .fla file, which it is currently, it will generate a version of the image-text for the letter Z that is displayed in the project. If you are interested in experimenting with some of the values in the .as file, I would suggest editing some or all of the values assigned to the following variables:

1. The numerical values for *r* and *d* in lines 22 and 23
2. An alternative .jpg or .gif file in line 27—you will also need to include the alternative image file in the same folder as the .as file
3. The text assigned to *str* in line 30
4. The name of the font in line 52
5. The base-10 color value in line 54
6. The size of the font in lines 55 and/or 58

Explanations of some of the key concepts and strategies in the software program are included in the section of this essay titled "Hybridity, Cultural and Computational." The text values in Table 1 were assigned to the *str* variable to generate the image-texts:

A	The Alpha Kappa Alpha sorority's creed
B	"And they will hammer their swords into plowshares"
C	"praise the lord jesus christ"
D	"Dix 306"
E	"electric"
F	"Compra Ganga Mercado"
G	Lincoln's "Gettysburg Address"
H	"hamsahamesh"
I	"Inmates Working"
J	"Joel Lane Father of Wake County"
K	"Doughnuts and Coffee Since 1937"
L	"Lake Johnson"

M	Excerpt from King's "I Have a Dream" speech
N	"IT PAYS TO BE WELL GROOMED"
O	A random selection of characters
P	"Deliciosas Pupusas y Tacos"
Q	"SERVING CAROLINA STYLE BARBEQUE SINCE 1938"
R	"Night Train/Miami, Florida/Atlanta, Georgia/Raleigh, North Carolina"
S	Lyrics from a song: "The Lord knows my name/I am not forgotten"
T	"shalom"
U	"26.2"
V	"whiRly GiG"
W	"WWI WWII"
X	"A 16oz cup of coffee with three shots of espresso"
Y	"Yard of the Month"
Z	"Bridezilla Kirby Derby"

Table 1. Text values assigned to the variable *str*.

3. Hybridity, Cultural and Computational

3.1. Greater Than the Sum of Its Parts

Within the past decade in composition studies, the term *hybridity* has been used to describe the semiotic dynamic between visual and verbal modes of communication in digital media. For example, writing about the Web in "Understanding Visual Rhetoric in Digital Writing Environments", Mary E. Hocks (2003) explained, "The hybridity of the Web medium refers to the interplay between the visual and verbal in one constructed, heterogeneous semiotic space" (p. 637). The recognition of the visual-verbal interplay has led to calls by compositionists to develop new approaches that foreground the interplay between the two modes. In his article titled "Visualizing English," Craig Stroupe (2000) characterized his approach as hybrid due to its focus on the dialogic interplay about which Hocks and others have written. In the following quote, he alluded to hybridity as a "double-voiced" rhetoric, which is one in which the "literacies" of verbal and visual cultures are intertwined:

> The more hybrid approach of a visualized English would describe instead the potential for dialogically constitutive relations between words and images—in a larger sense, between the literacies of verbal and visual cultures—which can function as a singly intended, if double-voiced, rhetoric. (p. 609)

An important and related point about hybridity as defined here is that the emergent forms of textuality arising from the interplay of image and word is greater than the sum of its parts. In other words, the visual and linguistic aspects of hybrid texts cannot be studied separately. Kristie Fleckenstein (2003) emphasized this point in her book, *Embodied Literacies*, when she wrote, "Although it has been necessary to separate image and word in order to understand the logics that constitute

them, doing so implies that imagery is somehow separable from language. Neither image nor word exists without the other" (p. 30). Echoing Stroupe's metaphor of a "double-voiced" rhetoric, Fleckenstein concluded, "Infused with double logics and double meaning, image and word are mutually constitutive, mutually creative" (p. 30). A comparison of the two figures below demonstrates the extent to which hybridity is compromised when visual and textual data are separated. Figure 7 is a detail from the image-text for the letter Z.

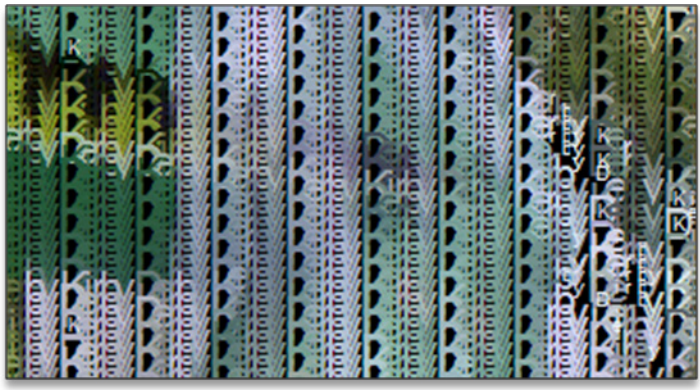

Figure 7. Detail from the image-text for the letter Z

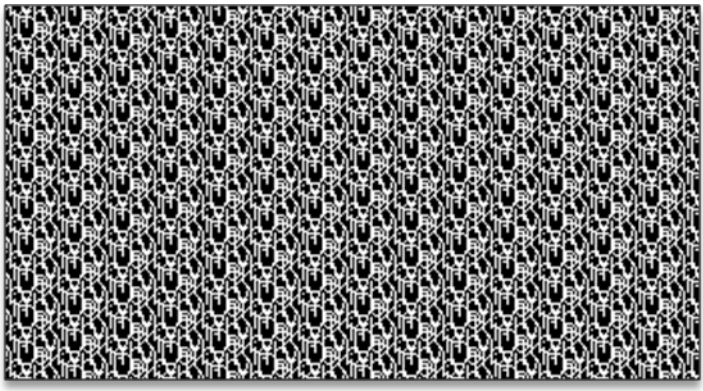

Figure 8. Detail from the image-text for the letter Z with some of the visual data removed.

Borrowing from Gunther Kress and Theo van Leeuwen's (1996) methods of visual analysis in their book titled *Reading Images*, the complex vectors of color and line in the detail are the basis for both narrative and conceptual readings. Vertical textures running up and down the detail imply action, which can be the basis for a narrative reading. By contrast, the line of text running diagonally across the detail can be valued as a conceptual marker of relationality. Compared to the original image, which, based on Kress and van Leeuwen's terms, I define as a "non-transactional" action, the hybrid revision of that image explodes with meanings that are both narratological and conceptual.

Figure 8 is a detail from the same section of the same image-text. The difference is that some of the visual data has been turned off in the software program. The two words, "Kirby Derby," iterate across and down the screen, but the complex vectors of narrative and conceptual meaning are missing. In their place, we have a black-and-white image that reads like static on a television screen. Based on this comparison, I agree with Fleckenstein and other scholars who argue that hybrid meanings must be approached with a method that preserves the delicate, dialogic interplay between image and text.

3.2. "Deeper into the Machine"

In her article titled "Deeper into the Machine: Learning to Speak Digital," N. Katherine Hayles (2002) called on compositionists to develop new "critical vocabularies" derived from the technical dimensions of the media about which they are writing: "This new critical vocabulary will recognize the interplay of natural language with machine code; it will not stay only at the screen but will consider as well the processes generating that surface" (p. 373). Due to the computational basis of the image-texts in this project, her proposed approach is compelling. There are numerous hybrid effects generated computationally that would not show up in most studies of hybridity because they would not be based on a reading of the code. Even Gunther Kress and Theo van Leeuwen's (1996) extensive methodology for analyzing visual design in books like *Reading Images* would miss the materiality of the medium, which, in this case, is AS3.

Based on Hayles' call for new terms, the following summary of the code in Hybridizer.as is meant to offer compositionists a few more terms on which to base an analysis of image-texts in this project. The explanation is divided in two parts based on Lev Manovich's (2001) principles of *numerical representation* and *transcoding*. In the first section, I describe the numerical bases of strings and bitmaps and explain their roles in Hybridizer.as. In the second section, which is also divided in two, I focus on the process of transcoding, explaining how it is the computational basis of hybridity in this project. In the first sub-section, the role of the two *for loops* are described. Following that, I focus on the ways in which the transcoded data is written to the textfields that constitute the image-text on the screen.

The explanations assume a certain degree of familiarity with basic programming concepts. Since there are so many tutorials on the Web that will explain these concepts, I glossed over them. Even if the following explanations are difficult to follow, I hope that you will take my advice (in the opening section of this essay) and experiment with the values assigned to some of the key variables.

3.3. Explanation of the Code

The algorithmic core of Hybridizer.as (see Table 2) is constituted by two *for loops* that simultaneously iterate through every other value in the bitmap graphic and every value in the string, *str*. Since the length of the string is significantly shorter than the distributed sample of visual data in a 300 x 200 pixel bitmap graphic, the values in *str* are reused dozens, hundreds, and sometimes thousands of times in order to generate an image-text.

The values from the bitmap and the string are transcoded to 15,000 individual

textfields. Each time the first *for loop* executes the second *for loop*, the second one adds 150 textfields along an x axis on the screen. The position of the textfields on the y axis are derived from the first *for loop*. The color, font, font size, and x-y position on the screen are derived from the visual data. Each textfield contains a single letter from *str*.

```
1   package {
2
3       import flash.display.Bitmap;
4       import flash.display.BitmapData;
5       import flash.display.Loader;
6       import flash.display.Sprite;
7       import flash.net.URLRequest;
8       import flash.net.URLLoader;
9       import flash.events.Event;
10      import flash.text.TextField;
11      import flash.text.TextFormat;
12
13      public class Hybridizer extends Sprite {
14
15          private var l:Loader = new Loader();
16          private var w:Number;
17          private var h:Number;
18          private var bitmapData:BitmapData;
19          private var bitmap:Bitmap;
20          private var counter:Number;
21          private var str:String;
22          private var r:Number = 2;
23          private var d:Number = 3;
24
25          public function Hybridizer() {
26
27              l.load(new URLRequest("Z.jpg"));
28              l.contentLoaderInfo.addEventListener(Event.COMPLETE, generateHybrid-
Text);
29
30              str = "Kirby Derby";
31
32          }
33
34          private function generateHybridText(e:Event=null):void {
35
36              counter = 0;
37
38              w = l.width;
39              h = l.height;
40
41              bitmapData = new BitmapData(w,h);
42              bitmap = new Bitmap(bitmapData);
43
44              bitmapData.draw(l);
45
46              for(var j:int = 0; j < bitmap.height; j += r) {
47
48                  for(var i:int = 0; i < bitmap.width; i += r) {
```

```
49
50                      var tf:TextFormat = new TextFormat();
51                      tf.color = bitmapData.getPixel(i,j);
52                      tf.font = "Arial";
53
54                      if(bitmapData.getPixel(i,j) <= 15000000) {
55                          tf.size = 16;
56                      }
57                      else {
58                          tf.size = 8;
59                      }
60
61                      var txt:TextField = new TextField();
62                      txt.x = i * d;
63                      txt.y = j * d;
64                      txt.text = str.charAt(counter);
65
66                      if(counter >= str.length) {
67                          counter = 0;
68                      }
69                      else {
70                          counter++;;
71                      }
72                      txt.setTextFormat(tf);
73                      txt.autoSize = "center";
74
75                      addChild(txt);
76                      }
77                  }
78              }
79          }
80      }
```

Table 2. Hybridizer.as

3.4. Numerical Representation of Strings and Bitmaps

In *The Language of New Media*, Lev Manovich (2001) listed numerical representation and transcoding as his first and fifth principles of new media. Since Hybridizer.as is, for all intents and purposes, a transcoding machine, understanding these two principles is essential. Manovich introduced numerical representation as follows: "All new media objects, whether created from scratch on computers or converted from analog media sources, are composed of digital code; they are numerical representations" (p. 27). On the screen, a word may be valued culturally as a verbal entity, but computationally it is both defined and manipulated numerically. Likewise, on the screen an image may be treated as a nonverbal form of communication, but in a programming language such as AS3, both the image and the word are numerical.

Although all new media objects are numerically defined, they are not all represented or accessed in the same ways. To explain what I mean, I'll contrast the ways in which text strings and bitmap graphics are represented numerically in AS3, since these are the objects on which Hybridizer.as is based. In *ActionScript 3.0 Bible*, Roger Braunstein, Mims Wright, and Joshua Noble (2008) introduced the string object as follows: "A string...can be any piece of textual data from a letter

to a word to several paragraphs" (p. 123). When a variable is typed as a string, it can be assigned a series of alphanumerical values. In line 30, the variable *str* is a string and its value is "Kirby Derby."

private var str = "	K	i	r	b	y		D	e	r	b	y	";
	0	1	2	3	4	5	6	7	8	9	10	

Table 3. A numerical representation of the alphanumerical values assigned to the string *str* in line 30 of Hybiridzer.as

Strings are implicitly represented by a range of numbers beginning with 0. Table 3 is an illustration of the way in which *str* is represented numerically. The string in Line 30, which is composed of a total of 11 spaces or slots, is represented by the range 0–10.

The properties and methods that let me access and change the values in this string are numerically based. In line 64, the *charAt()* method is used to identify the character in a specific numerical slot in a string. For example, if I add the command *trace (charAt(9))* in line 31, the value "b" will be displayed in the Output Panel. In Line 64, a (numerically typed) variable named *counter* stands in for a specific number. In line 36, it is assigned initially a value of 9, so the initial value of *charAt(counter)* is "K." But the value of *counter* will change as the program executes, so the value of *charAt(counter)* will change with each iteration of the program. The use of *counter* is related to transcoding, so I'll delve more deeply into its role in the program shortly.

Whereas strings are one-dimensional databases comprised of slots of individual textual characters, bitmaps are two-dimensional objects comprised of individual pixels of data. Not unlike the *charAt()* method that is used to identify the textual character in a specific numerical slot, the *getPixel()* method, which is used in lines 51 and 54, lets a programmer extract the color of a specific pixel as a numerical value. The bitmap graphic, Z.jpg, which is listed in Line 27, has a width of 300 and a height of 200, which means that there are 60,000 pixels in the two-dimensional grid (see Fig. 10). In order to identify the color assigned to a specific pixel, two numbers are written (separated by a comma) in between the parentheses of the *getPixel()* method. The two numbers represent the x–y location of the pixel in the grid of the bitmap graphic. For example, if I want to access the 50,780th pixel in the graphic in Figure 9, I write *getPixel(80,170)*. The value that will be returned is 8718412, which is the base-10 (decimal) version of the base-16 (hexadecimal) number 85084C, which is a dark, maroon-like red. Like the above-mentioned *counter* variable, which is used to assign a change set of numerical values to the *charAt()* method, the variables *i* and *j*, which are used in conjunction with the *getPixel()* method in lines 51 and 54, are used to access numerous color values within the bitmap for Z.jpg.

In sum, numerical representation is an essential characteristic of new media objects, but no two objects or data types are represented in the exact same way. Knowing how each type of object is represented numerically and how those representations are accessed and manipulated is essential.

Figure 9. An illustration of the way in which a bitmap graphic is represented numerically and its pixels can be accessed individually with the getPixel() method

3.5. Transcoding: For Loops, TextFields, and TextFormat

Manovich (2001) listed transcoding as his fifth (and final) principle of new media. It is a process that involves reassigning some of the values associated with one object to another one. Manovich introduced it as follows: "In new media lingo, to 'transcode' something is to translate it into another format" (p. 45). He concluded by characterizing it as "the most substantial consequence of the computerization of media" (p. 45). Hybridizer.as is a visual-verbal transcoding machine. In Hybridizer.as, the processes related to transcoding are found within the two *for loops*. In general, loops are used to execute one or more lines of code a specified number of times. Looping structures are an essential structure in every programming language. Their importance cannot be underestimated. In fact, Manovich claimed that the loop "gave birth" to programming (and cinema):

> the loop gave birth not only to cinema but also to computer programming. Programming involves altering the linear flow of data through control structures, such as "if/then" and "repeat/while"; the loop is the most elementary of these control structures. (p. 317)

One of the basic uses of the two *for loops* is to iterate through the rows and columns (or x and y axes) of the bitmap graphic. The loops are declared in lines 46 and 48. Figure 10 is an illustration of the ways in which the two loops interact. The largest circle represents the *for loop* in line 46. With each iteration, it executes the code between the { curly braces }, which is the second *for loop* in line 48. Like

two interlocked gears, each time the first gear is executed the second gear is triggered. The two *for loops* work together to iterate through a sample of the entire two-dimensional grid of pixels.

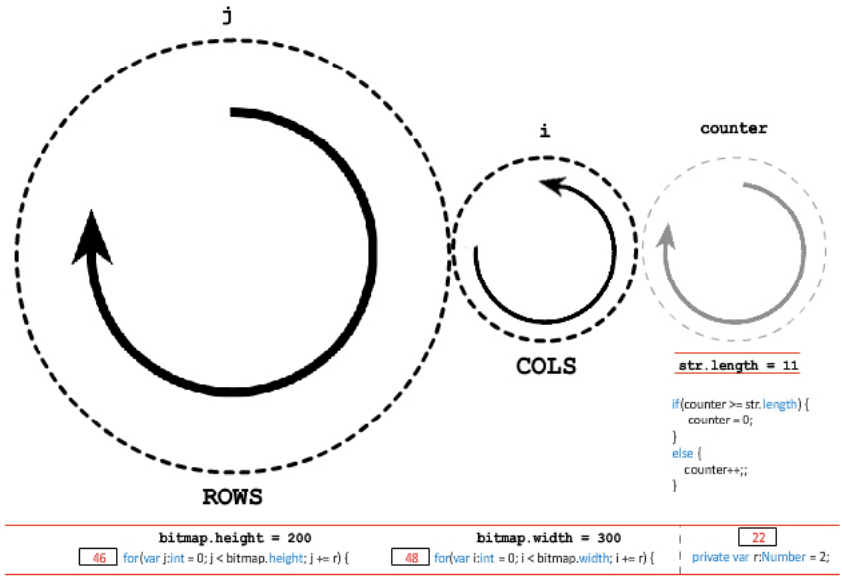

Figure 10. An illustration of the looping structures that transcode the visual data in order to determine the color, size, and placement of the text on the screen (zoom to read type).

At several points in this essay, I allude to the ways in which Hybridizer.as uses a distributed sample of visual data to generate the image-texts. The basis for the distribution is directly related to the variable *r*. In line 22, *r* is typed as a Number and assigned a value of 2. Lines 46 and 48, the value of *r* is used to cause both *for loops* to sample every other pixel in every other row in the bitmap graphic for a total of 15,000 pixels in 100 rows.

The third, smallest circle in Figure 10 represents the *counter* variable. The *counter* variable is used to iterate from 0 – *str.length*, which is 11. Since the overall length of *str* is significantly less than the number of pixels in the bitmap graphic, an if-else conditional is written in lines 66–71. The reason for the if-else conditional is to safeguard that the value of *counter* does not exceed the length of *str*. In other words, when *counter* has a numerical value that is >= 11, it is reassigned the value of 0. The effect of this if-else conditional is to rewrite the value “Kirby Derby” over and over again until the two *for loops* have reached the end of the last column in the last row of the bitmap graphic.

In AS3, strings are not display objects, which means that their values cannot be displayed directly on the screen. Two more types of text objects are needed to display or publish the transcoded values. They are the textfield and textformat objects. In the second *for loop*, the transcoded visual and verbal data is written to 15,000 individual textfields, each of which contains a single letter from *str*. In the following order, each textfield is assigned a color, font size, and, in conjunction with

the first *for loop*, a position within the x-y coordinate system on the screen. The color is applied to the textformat object in line 51, which is subsequently applied to each textfield in line 72. The size of the font is assigned in lines 54–59. The letter is assigned to the textfield in line 64.

4. Conclusion: Aleatoric Effects

In Figures 11 and 12, unexpected events of hybridity were generated by the chance combination of visual and verbal data related to pixel color, pixel location, font size, font type, and even the specific number of words in a text string. In Figure 11, font size, color, and placement on the screen contribute to a three-dimensional look in the middle of the detail. In Figure 12, these same factors contributed to the blurred effect in the top-left of the detail. These effects were not anticipated, but they are caused by the software program that generated them. Chance effects such as these are a relatively common occurrence when programmers experiment computationally.

Figures 11 and 12. Details from the letters M and W

While these effects were not intended, I contend that they are valuable as a form of aleatoric writing studies based in computational new media. Aleatoric writing has had a small but influential role in avant-garde prose and poetry in the twentieth century. It has also been an important characteristic of the experiments in writing associated with the Oulipo. In exploring numerate and literate forms of writing, the Oulipian scholars, writers, and mathematicians were obvious precursors for compositionists who use programming languages to develop new forms of hybrid and multimodal texts and visual rhetorical appeal.

An analysis of the on-screen effects of computational processes does not adequately describe the interplay between literate and numerate forms of writing. Whereas scholars like Hocks (2003) and Stroupe (2000) called on compositionists to respect the unique and delicate visual-verbal dialogue that is the basis of hybrid texts, the role of choice introduces an additional layer of concern. For compositionists analyzing hybridity in computational new media, the interplay between text and code ought to be represented. Especially considering that hybridity is generated programmatically, an analytical approach that delves deep enough into the machine to reconcile what Manovich (2001) would characterize as a composite view of the cultural and computational layers of new media objects is essential.

References

Braunstein, Roger; Wright, Mims H.; & Noble, Joshua J. (2008). *ActionScript 3.0 Bible*. Indianapolis, IL: Wiley.

Brown, James & the Famous Flames. (2004). Night train. *Live at the Apollo 1962*. Polydor.

De Certeau, Michel. (1985). Practices of space. In Marshall Blonsky (Ed.), *On signs* (pp. 122–145). Baltimore: The Johns Hopkins University Press.

Dawes, Brendan. (2006). *Analog in/digital out: Brendan Dawes on interaction design*. Berkeley, CA: New Riders Press.

Fleckenstein, Kristie. (2003). *Embodied literacies: Imageword and a poetics of teaching.* Carbondale, IL: Southern Illinois University Press.

Hayles, N. Katherine. (2002). Deeper into the machine: Learning to speak digital. *Computers and Composition, 19*(4), 371–386.

Hocks, Mary E. (2003). Understanding visual rhetoric in digital writing environments. *College Composition and Communication*, *54*(4), 629–656.

Hodge, Stephen; Smith, Phil; Persighetti, Simon; Turner, Cathy [Authors]; & Weaver, Tony [Illustrator]. (2003). *An Exeter mis-guide*. Exeter, England: Wrights & Sites.

Johnson, Stephen T. (1995). *Alphabet city*. New York: Puffin Books.

Kress, Gunther, & van Leeuwen, Theo. (1996). *Reading images: The grammar of visual design.* New York: Routledge.

Lincoln, Abraham. (1863/2005). The Gettysburg address. *The library of Congress*. Retrieved October 27, 2009, from http://www.loc.gov/exhibits/gadd/images/Gettysburg-2.jpg

Manovich, Lev. (2001). *The language of new media*. Cambridge, MA: The MIT Press.

Reynolds, Nedra. (2006). *Geographies of writing: Inhabiting places and encountering difference*. Carbondale, IL: Southern Illinois University Press.

Scurti, Franck. (2003). *Topo-Typographie*. Exhibit: GNS, Global Navigation System. Paris, France: Palais de Tokyo: Site de Creation Contemporaine.

Sirc, Geoffrey. (2002). Writing classroom as A&P parking lot. *English composition as happening* (pp. 185–233). Logan, UT: Utah State University Press.

Stroupe, Craig. (2000). Visualizing English: Recognizing the hybrid literacy of visual and verbal authorship on the Web. *College English, 62*(5), 607–32.

Trimbur, John. (2002). Delivering the message: Typography and the materiality of writing. In Gary A. Olson (Ed.) *Rhetoric and composition as intellectual work* (pp. 188–202). Carbondale, IL: Southern Illinois University Press.