Computational Models for Expressive Dimensional Typography

Peter Sungil Cho

S.B. Mechanical Engineering, Design, and Computation
Massachusetts Institute of Technology
June 1997

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences at the
Massachusetts Institute of Technology
June 1999

Copyright Massachusetts Institute of Technology, 1999
All rights reserved

Author

Peter Cho
Program of Media Arts and Sciences
May 7, 1999

Certified by

John Maeda
Sony Career Development Professor of Media Arts & Sciences
Assistant Professor of Design and Computation
Thesis Supervisor

Stephen A. Benton
Chair, Departmental Committee on Graduate Students
Program of Media Arts and Sciences

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
JUN 1 4 1999
LIBRARIES

ROTCH

Computational Models for Expressive Dimensional Typography

Peter Sungil Cho

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on May 7, 1999
in partial fulfillment of the requirements for the degree of
Master of Science
in Media Arts and Sciences

Abstract

This thesis research explores the prospect of typographic forms, based on custom computational models, which can be faithfully realized only in a three-dimensional, interactive environment. These new models allow for manipulation of letterform attributes including visual display, scale, two-dimensional structure and three-dimensional sculptural form. In this research, each computational model must accommodate the variation in letter shapes, while trying to balance functional flexibility with the beauty and legibility of fine typography. In most cases, this thesis work approaches typography at the level of a single letter, looking at ways we can build living, expressive textual environments on the computer display.

Thesis Supervisor: John Maeda
Title: Sony Career Development Professor
of Media Arts & Sciences
Assistant Professor of Design and Computation

Computational Models for Expressive Dimensional Typography

Peter Sungil Cho

The following people served as readers for this thesis.

Thesis Reader

Daniel Boyarski
School of Design
Carnegie Mellon University

Thesis Reader

Matthew Carter
Carter & Cone Inc.

Acknowledgements

Many thanks go to the people who have helped me during studies at the Media Lab and in the preparation of this thesis.

My friends in ACG, present and past: Ben and Elise, for making this year so much fun. Golan, for taking issue with my interpolation. Bill and Rich, for living the struggle. Tom, for being Tom. Dave, Reed, Matt, Chloe, Jared, Jocelyn, Catfoo, Yi, and Jessica, for sharing in this experience called ACG.

Phillip and Phoebe, for years of friendship and encouragement.

Glorianna Davenport, for introducing me to the Media Lab and the world of JBW, and Angelynn Grant, for seeing the typographer inside me.

My thesis readers, Matthew Carter and Daniel Boyarski, for their valuable insights and historical knowledge.

Katya, for some helpful last-minute thesis suggestions.

Professor John Maeda, for your consistency of vision. Thanks for teaching me so much. I am forever indebted.

Finally, Mom, Dad, and Austin, for your caring, support, and concern. I hope to make you proud.

Table of Contents

1 **Introduction**

Motivation *11*
Why is this thesis being done at the Media Lab? *14*
Thesis structure *15*

2 **Background**

Computational models for type *17*
Digital typography as experience *21*
Type in motion *23*
Sculptural letterforms *26*
Dimensional typography in print *28*

3 **Experiments**

Pliant type *31*
Type Me, Type Me Not *33*
Typeractive *39*
Stuttgart 21 *41*
Nutexts *42*
Letterspace *44*
Voronoi type *48*
Message: Stop Thinking So Flat *50*
Alphabet zoo *52*
ACG logo movie *55*

4 Analysis

Mechanics: Bits & pieces *59*
Categorization of experiments *63*

5 Thoughts on type

Letters as symbols *67*
On computational models *69*
Type can be a crutch *71*

6 Concluding remarks

A design philosophy *73*
What is this all for? *74*

A Implementation notes *79*

References *83*

1 Introduction

1.1 Motivation

We are surrounded by technology. Smart cars, smart homes, everyday computing not only at our fingertips but also woven into our clothing and our everyday life – this is the future proclaimed by many technological visionaries (Gershenfeld). As technologists make this computing power more accessible, some of us wonder, when will it be beautiful? The engineers at Microsoft can put a dancing paper clip on your computer desktop, and we ask, is that right? is that intelligent? is that a good design solution? As art directors and engineers harness the capabilities of high-end graphics stations to pump out frame after frame for big Hollywood blockbusters, the movie-going audience can say, amazing! breathtaking! fantastic! But another question arises, where can we hear the voices of the individual artists and designers in this emerging hybrid field of computational visual aesthetics?

We see some creative work on the Web, some personal stories and personal visions of interactive visual experiences. For a few years it looked like CD-ROM might be a viable medium, until corporations realized it was difficult to market and distribute, and not necessarily a money-making format. We also see amazing work in the genre of computer games: immersive, cinematic gaming experiences, made possible by those little polygon-crunching boxes labeled 'Sega' and 'Sony.' These are the emerging venues for a dialogue between art and technology.

We are at a point in history when the fields of art and design are being transformed by technology. In many cases, technology's main influence comes from a new tool which simply makes the work process faster. In some cases, the specific features and limitations of new technology can direct the creative process. Asked in 1989 about how new tools influenced her design work, graphic designer April Greiman, who developed a computer-influenced collage style in the 1980s, said:

> What I really miss now, are the great accidents that happened when I first started working on the Macintosh four years ago. At that time, the Macintosh threw me into an area where I wasn't so much in control anymore. I could do things that I wasn't able to do by hand. Accidents, messy things, kept happening. I'd use the wrong keyboard command or the mouse would get stuck, and these things would start happening, opening up a whole new road of possibilities that hadn't been heavily trod upon by other designers.
> (Emigre 11, from http://www.emigre.com/CE11.html)

As a tool, the computer can work its subtle influence in this way, guiding the creative process by making some actions easy, others frustratingly difficult. But in a few cases, technology gives the artist or designer an opportunity to create something impossible with conventional methods. Some of the current work along these lines, such as photograph mosaics which, from a distance, reveal a larger image, represents a twist, or gimmick, on an established field in the visual arts. And often, when it is the technologist who stakes claim to this new high-tech method, he or she, for lack of an artistic vision, just doesn't do it right. Other work, however, represents an informed attempt to use technology from the ground up to create an unexpected, innovative, and intelligent visual experience. When this happens, the computer becomes something more than a tool; it becomes a medium for creativity.

This thesis research attempts to expand the field of typography through an informed use of computation. Typography, or the arrangement of text elements on a given format, is an established art, practiced and explored through book, publication, and poster design throughout the past few hundred years, and in film title and motion graphics design in the past few decades. Type design, the design of letterforms and type faces, is an art with a tradition which extends to the beginning of written communication. With the advent of the web, digital media, and the rise of the information age, we have seen an explosion in "new" typography: letterforms that crack, fly, scream, decay, or otherwise animate as they try to communicate their message. Now is the time to evaluate the computational models which are used to represent the shapes we read as letters on the computer display and to look at new expressive ways these models can be manipulated.

By building custom computational representations for typography, we can examine new ways letterforms can exist in a virtual three-dimensional environment. We can explore new ways that typeforms can be displayed and new behaviors they can take on. We can also see how these custom representations affect the tone or emotional impact of a message and even look at new kinds of messages which are made possible only through a new way of representing the type.

This thesis research is motivated by several personal concerns and interests. I am a designer at heart, and I am the one who asks, why is that paper clip dancing in the corner of my screen, why did some designer-engineer make those decisions, and how could I have done it better? I am motivated to find the elegant design solution that is suited to the problem, the audience, and the medium. This clean design solution becomes elusive when the medium is as open-ended as the space behind the computer screen. It becomes harder to grasp when the audience is the hard-to-please advisor, colleagues, and self. And still further out of reach when the problem becomes simply, as it has in this research, to innovate.

Over the course of three years, I have shown my projects to diverse visitors to the Media Lab. Some of these visitors are design professionals who are awed by the technology; some are managers or engineers or businessmen who are unimpressed or genuinely puzzled by the work in the Aesthetics and Computation Group.

Regardless of who is visiting, the question always comes up: what is this good for? One answer we give is that the lessons learned through the research into abstract interactive and three-dimensional form can give us deeper insights into the visual-computational medium. The concepts we are exploring can be applied to information and interaction design problems as they become more complex. While this is true, my personal answer to 'what is this good for?' is, it's good for me. This research has given me an opportunity to explore the medium and see how I can express myself, a chance just to make new things and see how they come off.

1.2 Why is this thesis being done at the Media Lab?

In the past few years, educational programs in New Media, Multimedia, and Motion Graphics design have cropped up within the design departments of many universities in the U.S. and abroad. These programs for the most part teach students how to use the tools of the profession, the commercial software packages which allow for click-and-drag interactivity, key framing animation, and three-dimensional modeling. While these tools become more sophisticated with each new software release, the focus on tools neglects the power of working with lower-level computation. Reed Kram describes the study of "new media design" this way:

> If one is studying furniture design, the student should build at least a single chair: cut the wood, stretch the leather, then sit on the chair and test how it works. The same can be said for the interactive design, media design, what have you: 'build your own.' (Kram, p. 17)

The Aesthetics and Computation Group at the MIT Media Laboratory, to our knowledge, is the only concentrated effort to expose computation as a medium for visual exploration within the context of traditional graphic design. Being in the ACG, on a more personal note, has given me a chance to exercise both sides of my brain, the technical and the artistic, and hopefully contribute something innovative to this emerging field. During my three years in the ACG, my research interests, even when prodded in different directions, have always come back to typography, a passion of mine since using my first Macintosh computer at age 12. In this thesis research, I explore expressive typographic design in the context of computation, which would only be possible in the MIT Media Lab's Aesthetics and Computation Group.

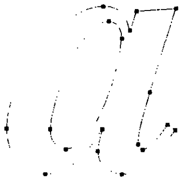
1.3 Thesis structure

The research presented here considers the prospect of typography based on custom underlying computational models which allow for the manipulation of two- and three-dimensional type forms, beginning at the level of the single character and moving up to the level of a word, a phrase, and beyond. Chapter 2 discusses the background for this work, including an overview of current models for type on the computer, a survey of dynamic typography examples in film and computer media, and a few examples of dimensional typography. Chapter 3 details and discusses the design experiments. In Chapter 4, I describe the technical bits and pieces involved in this thesis research and analyze the design experiments based on several criteria. Chapter 5 serves as a repository for thoughts on type. The final chapter expands this discussion to computational media in general in the context of this thesis research.

2 Background

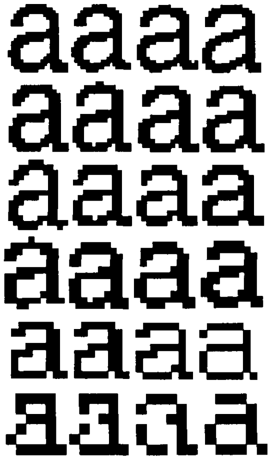
2.1 Computational models for type

Type has evolved throughout history, at times carved into clay, chiseled into stone, and cast into metal. For most computer users today, the most common use of type entails going into the font menu on the desktop to choose Times New Roman or Helvetica Bold. We use type every day without thinking about its underlying representation.



A TrueType outline

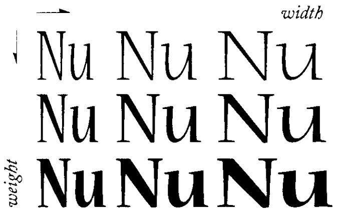
Several different computational models of typography have been introduced in the past few decades, including the two basic schemes we use today: Postscript Type 1 and TrueType. When Postscript, developed by engineers at Adobe, was introduced to the public in 1985, it revolutionized the printing industry because it allowed for a standardized scheme for describing all of the elements on a page. A single document could now go from computer screen to low-end laser printers or to high-end imagesetting machines.



The Helvetica 'a.' From bottom to top, increasing resolution. From left to right, increasingly savvy bitmap rasterizations. (from Hofstadter, p. 597)

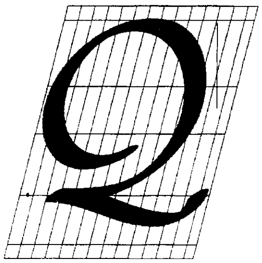
In both Postscript and TrueType, each character outline of a type font is represented as a list of two-dimensional data points connected by straight and curved segments. A type designer using these font description languages painstakingly positions the points comprising contours which are filled, creating a positive form. In general, these data points can be accessed by users of the typeface only through vector-based drawing software tools which allow the type to be converted to modifiable paths.

One issue which comes up when dealing with type on the computer is the screen display. When Postscript type fonts were first introduced, the onscreen representation existed as a separate file, called a bitmap. The bitmap rasterization, a pattern of pixels turned on or off, had to be "tweaked" by hand because programmatic bitmaps result in inconsistencies of character strokes and spaces between strokes. A technique of "hinting," giving fonts the built-in intelligence to adapt themselves pliably to raster grids, was developed for type fonts to correct inconsistencies of display at low resolutions. In the early 1990s, anti-aliased type, a means of substituting intermediary grays along edges to simulate smooth curves, made its introduction in work at the MIT Media Laboratory's Visible Language Workshop. Current Postscript and TrueType engines can now generate anti-aliased fonts in real time on the computer desktop.



The multiple master face Nueva with two axes: width and weight

While the Postscript-style method of defining type as a collection of outlines has proven successful, low-level type representation makes it difficult for certain types of manipulation because it has no additional information (aside from hinting) about the typeface or letters in general. Some efforts have been made to add some functionality to type on the computer beyond the single set of absolutely-defined characters. Multiple Master fonts, introduced by Adobe in 1991, suggests one means for adding a level of abstraction to Postscript fonts. In a Multiple Master font, a typeface is defined by a set of two or more outlines, representing the extremes of the visual representation, and the Multiple Master engine creates single "instances" by interpolating among the extremes.



A Metafont Q

A notable academic exploration in the field of computational type design was the Metafont project of the early 1980s, developed by programming guru Donald Knuth and collaborators at Stanford University. With Metafont, a user specifies a number of type parameters including x-height, width, stroke weight, and serif qualities, and the Metafont engine generates a font based on those characteristics. This scheme introduces the computer science theme of "abstraction" to type design. Since our concept of letterforms exists abstractly – the concept of a letter A exists above and beyond any specific A we might recognize – we should be able somehow to teach a computer to understand this higher-level abstraction. Along the way, we should also be able to teach the computer about serif and sans-serif, roman and italic, classical and modern. Theoretically, this should work, but in practice, teaching a computer about type is not such an easy problem, as Knuth discovered over the course of this nine-year project. Metafont, as it was finally released, uses pre-coded information about how a generic alphabet should look, then uses parameters to create a specific set of characters. It also includes a number of refinements, or hacks, introduced in the later stages of the project in order to make the character sets more acceptable by typographic standards. While the Metafont system is extensive and quite clever, it does have some drawbacks: the scope of typefaces the system can “spit out” is small; Metafont does not allow for personality in individual characters; and most importantly, the system perhaps suffers from the fact that Knuth, as he readily admits, is not a type designer.

As Postscript became recognized as a standard, and the personal computer popularized desktop publishing for the masses, several movements in graphic design embraced the new technologies and the crude capabilities of early software. Emigre, a type house and magazine founded by Zuzana Licko and Rudy VanderLans, represents a typographic movement spurred by the proliferation of the Apple Macintosh computer in the 1980s. Since the tools at first did not allow for high-resolution typography, Licko and VanderLans exploited the highly pixelated graphic nature of early Macintosh fonts and developed their own abstracted and geometrically ornate typefaces. Emigre states that they

> embraced the new digital medium. For them the computer soon became much more than just a design or production tool. From that point on, graphic design was never the same for us. Although it was a primitive tool during those first years, this computer made many new things possible. It was the Macintosh that made it economically possible to continue publishing. It also inspired us to design and manufacture original typefaces, an area previously dominated by only a few large type foundaries.
> (http://www.newcastle.edu.au/department/fad/ds/grad98/kylie/thirteen.htm)

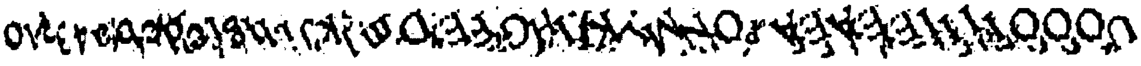
EMPEROR 8
UNIVERSAL 8
OAKLAND 8
EMIGRE 8

Early Emigre typefaces

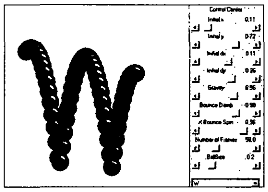
As new tools emerged, type on the computer could "do more" and the interest in type as high-tech image led to the technologically-motivated movement to obliterate type beyond recognition. The work of art director David Carson in *RayGun* and other trend-setting publications proclaimed "the end of type," or at least the end of type as we have grown accustomed to seeing it, with its use of typography to subvert the message.

Along the lines of Emigre's experimental letterforms, some attempts were made to encode computational variation into typefaces designed within the scope of the Postscript standard. Just van Rossum and Erik van Blokland's font Beowolf, for example, uses computational randomness to create slightly different instances of characters every time they are generated. Van Blokland's font Nimida, similarly, randomly degenerates its letterforms.

ABCDEFGabcdefghijklmnopqrstuvwxyz

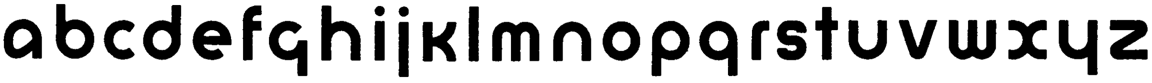


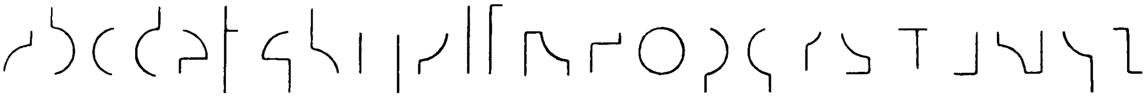
In the fall of 1997, Prof. John Maeda offered a course in Digital Typography at the MIT Media Lab. The course explored diverse issues relating to type on the computer including, though not limited to, different computational representations for type design. Tom White, a fellow graduate student in the Aesthetics and Computation Group, developed an alphabet in which letterforms were drawn by the path of a bouncing ball in a physical system. He designed the letters by assigning a dozen parameters, including starting position and velocity of the ball, the strength of gravity and the drag of the system, then set the ball loose. This is an example of a playful and truly unique computational model for type.



Tom White's font editor. Changing the nine parameters on the right affects the bouncing ball simulation.

White's bouncing ball typeface represents a recent example of a pattern which has been reflected throughout the history of type design: the rationalization of type. Over the years, type designers have repeatedly tried to develop rules for taming the diverse character set of typeforms. This trend, perhaps beginning in the early 1700s, when a project to design a typeface for the king of France suggested taking letterform proportions from an imposed eight by eight grid, was also reflected in the early part of this century, when modernist designers used a restricted repertoire of elementary shapes to compose typefaces. Examples of these efforts include Josef Albers' stencil typeface, and Herbert Bayer's Bauhaus "universal" face, both from 1925. The Polish designer Wladyslav Strzeminski designed a further rationalized font in 1931, drawing each letterform as a single unbroken line of right angles and the arcs of a single circle. We see this trend to rationalize type continuing in the digital medium with projects like Knuth's Metafont. Reducing a typeface to a set of simple parameters, as in White's font, can allow for new kinds of manipulation on the computer, such as letter-to-letter transitioning or controlled letterform deformation. Some of these capabilities, afforded by the computational medium, are explored in this thesis research.

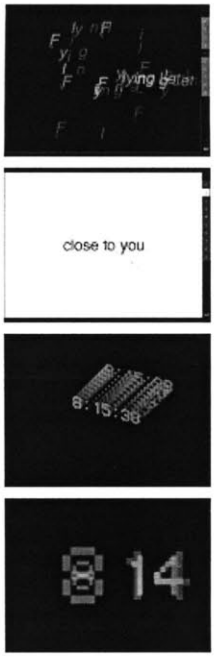




2.2 Digital typography as experience

While a useful tool for bringing typography to media such as print or video, the computer can also be a medium in itself for interactive experiences with texts. The web offers an opportunity for hypertext, documents which are linked to each other either linearly or non-linearly, and allows for multi-threaded story experiences. In hypertext, the computer is often used to make information more accessible: the reader can click on an unfamiliar word to find out more about it, or can search for a specific phrase among many documents. Beyond this kind of pragmatic functionality, the digital medium can also be an expressive one, through user interaction, computation, and visual feedback. A digital art field is emerging, in which the motive is not necessarily to provide access to information, but to create a compelling, computer-enabled, visual experience.

One of the pioneers in the computational art and design fields is John Maeda, whose work represents a tremendous inspiration for this thesis research. His work, including the Reactive Books series, develops the issue of "computational expression" within the context of basic interactions with the computer – the mouse, the microphone, the keyboard – which result in deceptively simple visual feedback. In several of the Reactive Books, namely *Flying Letters*, *12 o'clocks*, and *Tap, Type, Write*, he challenges our preconceptions of how the computer can react to and perform with the user and how typography can be an elemental part of that interaction. In *12 o'clocks* and *Tap, Type, Write*, Maeda uses typeforms designed within a small 5 by 7 pixel grid, exploiting our perception of this kind of type as early "computer type," but taking this device to a higher visual end through interaction and elegant graphic animations. His work suggests that a compelling visual experience on the computer does not have to employ high-end 3D graphics; computational art can rely on a set of two-dimensional shapes, a basic color palette, a mouse and keyboard, an understanding of temporal manipulation – and it can fit on a single floppy diskette.



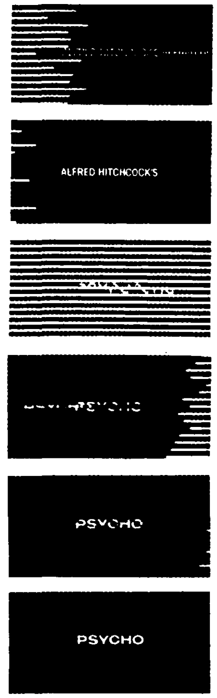
Stills from Maeda's Reactive Books. Top two images, Flying Letters; bottom, 12 o'clocks.

Jason Lewis, in his masters thesis project "Dynamic Poetry: Introductory Remarks to a Digital Medium" at the Royal College of Art, approaches typography from a content standpoint. His work features a series of experiments in interactive and animated typography set in the context of traditional poetry – with words representing concepts – and concrete poetry – in which words are also treated visually. In one notable experiment, WordNozzle, a user points a fire hose nozzle at a wall display, and the words of a poem spit out onto a wall display. In this piece, Lewis recasts a poem in the realm of performance to bring about a more immersive experience. He explores in several pieces how poetic content can be developed especially for the digital medium, remarking, "The need to write poetry which, from the first creative moment, is destined for a digital environment had been made increasingly clear." In Breeder: No Discussion, for instance, Lewis gives a word knowledge of its own location within a larger piece and combines with other sequential entities over time until the entire piece is reconstructed. While a bit crude from both a visual and technical standpoint, Lewis' work suggests an unchartered space where poetic content, visual design, and interactivity intersect.

2.3 Type in motion

Type appeared in a temporal medium before the computer ever came along, namely the medium of film. Designers and filmmakers have dealt with type in motion in different ways during the one hundred year-long history of cinema. In the early part of the twentieth century, type appeared out of necessity in the "inner-titles" of silent films, black cards detailing the captions in order to make the story explicit. In most films since that time, type appears as static, sequential placards at the opening title sequence, and as a rolling scroll at the end. In some films, such as "West Side Story" and "Delicatessen," the titles of the film become a more cinematic experience by employing found type, the temporal dimension coming in as the camera moves through the set. In other films, the type itself moves, sometimes layered on top of an image plane, sometimes in an imagined picture plane of its own.

Film title sequences emerged as its own "artform" in the late 1950s with American designer Saul Bass' graphic films-within-films. Working with directors Otto Preminger and Alfred Hitchcock, Bass developed vivid, animated opening title sequences



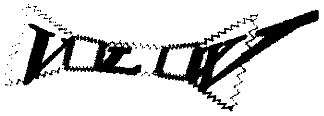
Saul Bass' title sequence for Psycho, 1960.

for "The Man with the Golden Arm" and "Vertigo," among others, which echo the emotional content of their films. In his opening titles for "Psycho," for instance, horizontal lines move across the picture to reveal sliced and broken letterforms, conveying uneasiness and alluding to the jarring violence of the shower scene.

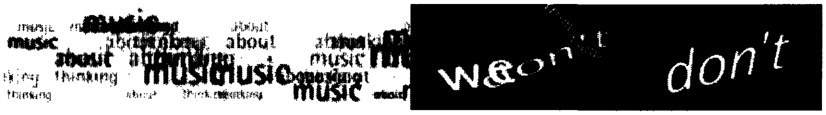
A notable film titles designer of this decade who has treated type in motion in an innovative way is Kyle Cooper. His design work for "Seven" (1995), a film which fueled a new wave of interest and activity in film titles design, entails an intensely crafted, multi-layered visual essay detailing close-ups of the serial killer's eccentric activities. Flickering hand-scrawled type becomes a textured layer in this sequence which highlights the uneasiness and suspense of the film. Cooper goes further with the use of type as visual texture, as popularized also in print media by David Carson and others, in the film title sequence for "Island of Dr. Moreau" (1996). In this sequence, Cooper applies the film's theme of genetic mutation to the typography – the titles' letterforms grow spikes and thorns at a blistering pace, reaching a point beyond recognition. His recent titles use analog and digital techniques to apply innovative visual effects to type as image, such as placing type underwater or under a magnifying glass.

In contrast to dynamic type in film, animated type on the computer has only begun to be explored. At the pragmatic end of this study is a technique called Rapid Serial Visual Presentation, or RSVP (Potter 1984, Mills et al. 1987). Research in RSVP suggests that readers can absorb text rapidly when the focal point is fixed and text is displayed serially, word by word. This method is motivated by an economic use of screen-space and an interest in presenting relatively large amounts of text without scrolling. The Kinetic Design Group at Carnegie Mellon University, under the direction of Suguru Ishizaki, explores the expressive capabilities of the RSVP method. In many of their design exercises, type displayed sequentially takes on qualities of a spoken voice including tone, emotion, and personality. In an exercise by Robert Pietri entitled "Phone Call," for example, a single text is given two different treatments of pacing, motion, and type size, to very different ends. In the first, the voice on the telephone line, humorous and offbeat, seems to be jumping for joy at the news that his mother-in-law has died. In the second, the voice's reaction is more of what one might expect: sad, tired, shocked.

Other formal research into expressive temporal typography came from the MIT Media Laboratory's Visible Language Workshop (1974–1994). The VLW, led by Muriel Cooper, pioneered the "information landscape," the use of the implied three-dimensional space of the computer screen to lay out functional and expressive texts. The work of David Small and Yin Yin Wong in particular contributed to the concept of typography living in an abstract 3D space. Small (1987) applied simulated Newtonian physics to typography, giving real-world physical attributes to type on the screen. Wong (1995) developed a framework for thinking about and designing temporal typography, along with some compelling examples. In her piece "Red Riding Hood," Wong presents the dialogue between the young girl and the wolf, masquerading as the grandmother, by using the space of the screen as a stage for the text of the two actors. She uses transparency and motion to suggest the actions and the appearances of the two characters. In 1996, Small and Wong collaborated to design a typographic animation for the Brain Opera, a Media Lab musical exhibition and performance. In this six-minute piece, type flies through the implied three-dimensional space of three large projected screens, choreographed in time with vocals. This piece effectively creates an immersive experience, combining



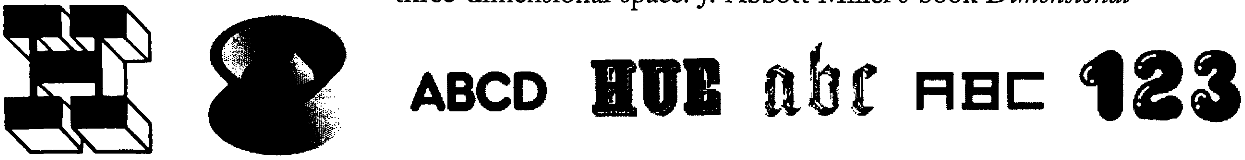
David Small's spring-based type manipulation



Two stills from the Brain Opera expressive type animation text with music. Also the work of Suguru Ishizaki (1996), and his comparison of typographic elements to dancers in a choreographed piece, contributed to a formal understanding of how to coordinate text in a complex, dynamic information graphic.

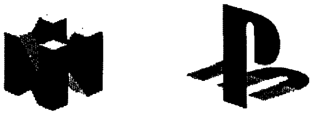
2.4 Sculptural letterforms

We see a few precedents for considering letterforms as objects in three-dimensional space. J. Abbott Miller's book *Dimensional Typography* gives a short historical overview of dimensional treatments of typography. He places three-dimensional renderings of letters into categories, including extrusion, rotation, tubing, shadowing, sewing, modular construction, and bloating. In *Dimensional Typography*, Miller continues, presenting case studies of dimensional letterforms in virtual environments. Many of these design studies present two-dimensional forms which are extruded and revolved; others treat the letter as a sheet which can be "warped" in three-dimensional space.



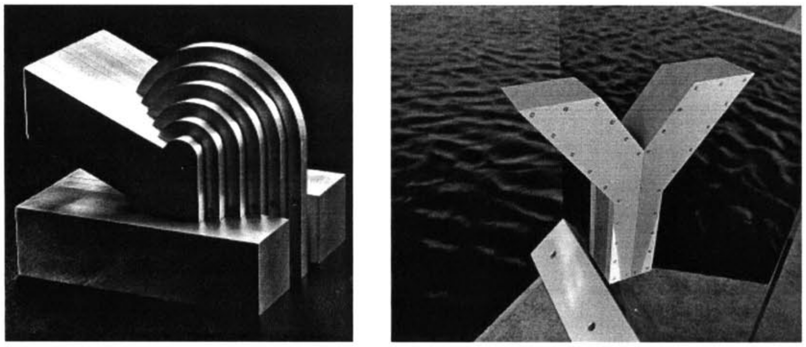
Examples of type featuring three-dimensional rendering, from Miller, 1996. From left to right: extrusion, rotation, tubing, shadowing, sewing, modular construction, bloating.

In the 1990s, we have seen a surge of 3D graphics in films and video games. One result of this visual onslaught is the emergence of graphic identity systems which imply three-dimensionality, often branding companies in high-technology industries. One good example of this trend are the logomarks for dueling Japanese game companies, Nintendo and Sony Playstation. Both logos use playful, brightly colored shapes to suggest 3D forms based on the initial letters of the company names. The logos both play with the dimensionality in subtle ways; the S of the Sony Playstation logo is the shadow of the P, while the Nintendo logo represents a rotating, sculptural N.



Competing sculptural logomarks: left, Nintendo; right, Sony Playstation

In a series of physical sculptures he designed in the 1980s, Takenobu Igarashi dissects, extrapolates, and abstracts the two-di-



Two of Igarashi's letterform sculptures. Left, R, 1983. Right, Y, 1987.

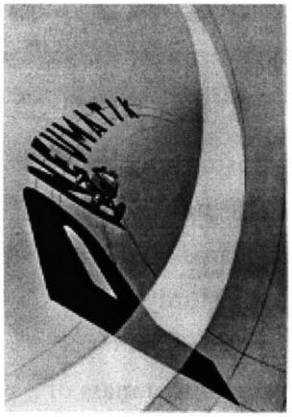
mensional letterform into three dimensions in surprising and playful ways. In one set of sculptures, letterforms become stair-stepping concentric aluminum forms punctuated by square beams. In another set, he examines the symmetry of letters, fixing sturdy, brass half-letterforms on polished aluminum plates to create a floating alphabet. In a third series, he folds flat steel letterforms into a mass of angles and geometrical planes. Igarashi delights in the abstracted quality of single letters, saying

> One of the charms of the Roman letter is its simple form. The wonderful thing is that it is created with the minimum number of elements; the standard structure is based on the circle, square and triangle which are the fundamentals of formation. I create complicated, three-dimensional works based upon this simple structure.
>
> The second charm of the Roman letter is its strength as a symbol. No other character is as popular in the world. From children to adults, regardless of whether they are Asian or European, people will embrace the familiarity of an alphabet sculpture.
>
> In most cases, the creation begins from the shape of the character in its two-dimensional form. When the two-dimensional form (plane) is completed, a three-dimensional form (cube) is constructed thereon in the same way an architect would. occasionally there are times however, when the shape of a character is that of a vertical plane which represents, what we call in architecture, an elevation. (Igarashi, p. 58)

Igarashi's work, in both sculpture and print, represents a creative mind who can move freely between two dimensional and three dimensional design, drawing from the best of both realms.

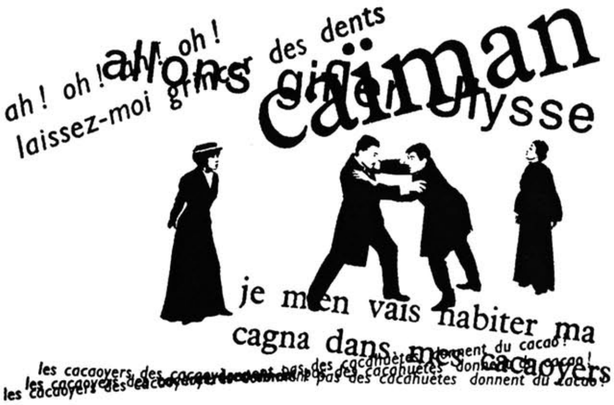
2.5 Dimensional typography in print

The discussion of typographic precedents influencing this thesis research would be incomplete without a mention of expressive and dimensional typography in print. Graphic designers have a tradition – predating the cathode ray tube – of treating the printed page as an implied three-dimensional space. Some designs draw from the way we see perspective in the physical world, displaying type in a dynamic and spatially-motivated manner. The Bauhaus artist Lazlo Moholy-Nagy, for instance, gives letterforms a sense of motion in a poster for Pneumatik tires by distorting the type and placing the letters on a plane other than the flat plane of the page.



Moholy-Nagy's Pneumatik poster, 1923

In his design of Eugene Ionesco's absurdist play *The Bald Soprano*, the French designer Robert Massin treats the page as a stage for type. Photographs and lines of text are orchestrated to represent the inflections of voice, awkward silences, and commotion of multiple voices visually. Massin uses type to suggest the characters' movements on stage, vocal expressions, and emotions. To achieve these type effects, Massin used a creative combination of optical and analog techniques, like stretching type which has been imprinted onto elastic surfaces.


ah ! oh ! ah ! oh !
allons
laissez-moi grincer des dents
caïman
Ulysse
je m'en vais habiter ma cagna dans mes cacaoyers
les cacaoyers des cacaoyères donnent pas des cacahuètes donnent du cacao !

Spread from Ionesco's La Cantatrice Chauve, designed by Robert Massin, 1964

In this thesis research, I consider typography on the computer in the context of the history of print, animated, and dimensional typography, along with a short history of precedents for developing type computationally. While my design experiments consider the research of computer scientists such as Knuth, developer of Metafont, I tend toward more abstract, expressive aspects of computational type.

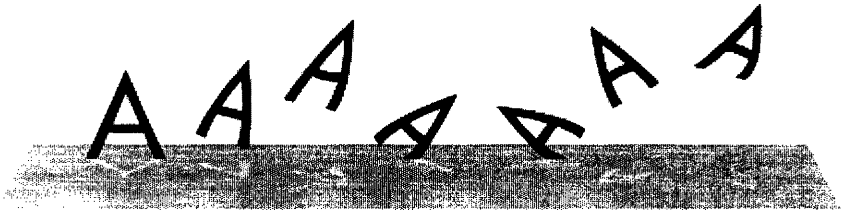
3 Design experiments

Within the scope of this thesis research, I have developed a series of design experiments exploring different aspects of expressive, dimensional, and computational typography. This section details these projects, beginning with letterform experiments conducted previous to this thesis project and ending with typographic pieces from recent months. I discuss the relevance of these experiments in the context of computational representation of type and type design in general, as well as the overall experience as a digital design piece. I also try to be critical of these pieces, pointing out problematic elements and suggesting further directions.

Further documentation of these design experiments is available on the Internet at http://www.media.mit.edu/~pcho/thesis.

3.1 Pliant type

The majority of these typographic experiments fall into three categories: shape manipulation of individual letterforms, transition between letters, and exploration of three-dimensional and two-dimensional space with typography. The earliest design experiments treat letters as malleable shapes which can move fluidly in time. These experiments in *pliant type* were conducted during the 1996–97 school year when I worked as an undergraduate researcher in the Aesthetics and Computation Group. While these pieces predate my thesis research, I am including them here because they introduce many of the ideas about expressive type I have been developing since that time.



Letter Dance animated 'A'

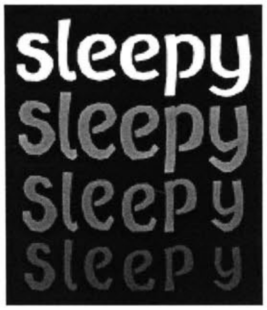
Letter Dance

In the first piece, Letter Dance, I designed a single letterform, the A, as a playful interactive element in a simple 3D space. As the user moves the mouse, the letter moves and changes shape, appearing to dance and smile. The apparent fluidness of the letterform results from a time difference between the user's mouse point and the updating of the letter's skeletal points. As the mouse input location changes, the top and sides of the A follow, but the midpoint of the A crossbar and the two bottom "legs" lag behind. One interesting point that makes the dancing A effective as a piece is that the viewer is eager to ascribe humanlike characteristics to the pliant letterform, both because the shape deforms fluidly and because the letter A is humanlike in appearance, with its two legs.

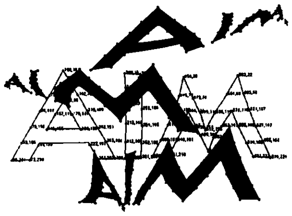
LMNOP

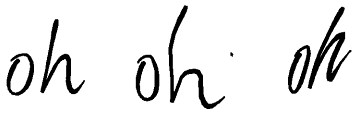
After working on the Letter Dance, I did a small study of letter-to-letter transitional forms. In this experiment, I constructed the letters L, M, N, O, and P (all of which can be drawn as a single, connected line) from a line comprising six segments, some straight, some curved. As the user moves the mouse across the five letters, the eleven control points interpolate linearly to create transitional shapes. Some of the transitions seem awkward and unexpected due to the programmatic computed interpolation. This small design experiment begs for a more fluid line, which would require a more sophisticated representation.

Sleepy, Aim, Oh



In several subsequent experiments, I explored shape deformation of letterforms at the level of a single word. In two pieces, Sleepy and Aim, the words are defined by outline points which are manipulated on the screen by a predetermined script or by the mouse. In both of these experiments, the points move toward or away from a single point in a way that gives a fluid quality to the letterform shape animations. Sleepy plays out as a time-based animation: from the starting position, the letters droop downward, then jump back two times, then finally drop slowly to a resting horizon. At this point, the letters slowly expand and contract while fading out to the background color. In Aim, the letters begin from a starting position and deform toward the cursor as the mouse button is pressed. On release, the letters float back into position. In a third design experiment, Oh, the letterforms are defined by a list of points that make up the skeleton of the letter shape, instead of the outer contour. I developed this skeletal type concept to make up for what I thought was a weakness with the outline representation – since we write letterforms as strokes, a stroke-based representation for type seems natural. Furthermore, when an outline representation is used, there is no intelligent way to change the weight of the letter, which would be desirable. In this type representation, a stroke width is designated at each control point to give the letters the effect of being drawn by a variable width brush. In the Oh piece, the points are manipulated in three different modes, taking in the mouse point as a parameter.



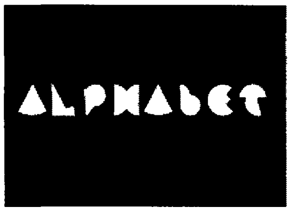


In these three experiments, my goal was to make the letter shapes "pliant" in an effort to give an expressive quality to type which goes beyond animating static letterforms. These simple and fairly crude examples serve to introduce some basic ideas about the representation of two-dimensional typeforms.

3.2 Type me, Type me not

In fall of 1997, I took part in Computational Typography, a course taught by Prof. John Maeda at the Media Laboratory. The problems assigned in this course examined different aspects of type on the computer: creative text "filters," innovative type representations, and connections between letterforms and the phonemes they represent. After completing the course, I combined two of my exercises from the class with a new type experiment to make a three-part piece called Type me, Type me not: Experiments in computational typography. This piece, running as a Java applet on the Web, received a Gold Award in the ID Magazine 1998 Interactive Media Review. The piece consists of three different modes, called A, B, and C, each featuring a different animated alphabet that is activated by pressing the keyboard. Part A deals with the way one letter transitions to another; part B examines the connection between a letter and the sound it makes; and C looks at the intersections between different letter shapes.

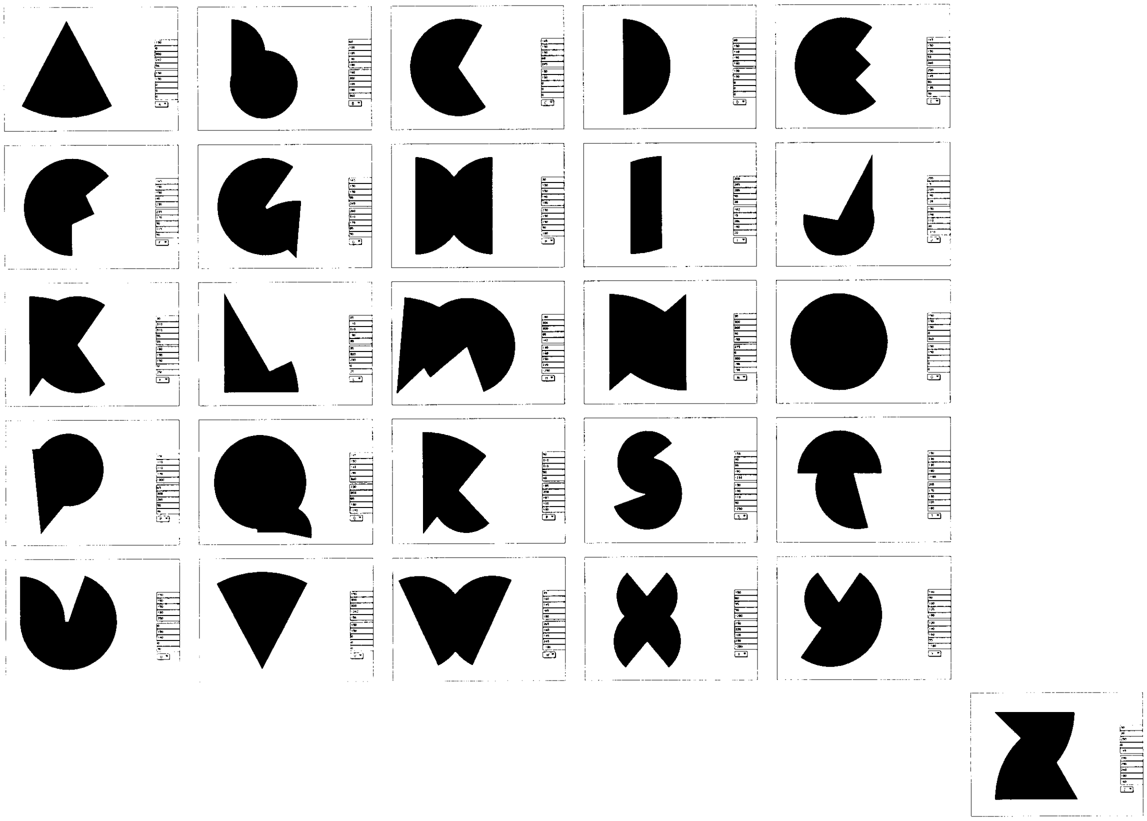
Type me not, A



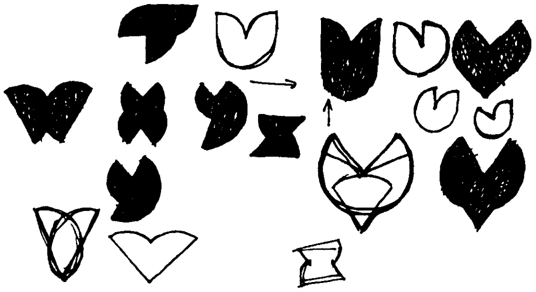
In part A, abstracted letterforms emerge and transition from one to the next as the user types into the keyboard. When the user presses the carriage return, the letterform on the screen spawns the previous succession of letters to form a word. Part A came from an assignment in the typography course asking us to develop a vector-based typeface that could be "blurred." I addressed this problem in the temporal dimension – in my solution, blurring occurs during the time in between, when one letter is transitioning into another.

In this type font, each letter is composed of two "pie-piece" shapes, or filled circular arc. A filled arc is a versatile shape: depending on the parameters, it can be a long wedge, a nearly-triangular wedge, a semicircle, a pacman shape, or a full circle. This repertoire of shapes allows for different shape characteristics, sharp, rounded, or rectilinear. Once I had decided on this rigid set of constraints to work with, I found some letters to be easy to design (O, H, D), while others presented a struggle. For example, once I had decided on a P, in which each of the filled arcs played essential roles, it took some time to find a R. In this case, I had to leave the context of the P and look to the K for inspiration. In general, I tried to avoid stroke-like usage of the pie pieces (as in L) and instead tried to use the shapes to "build up" the letter forms (as in N).

Where this piece comes alive is in the transitions between letters,



In Type me not, A, each letter is composed of two filled arc primitives. This extreme design constraint lends itself naturally to some letters, like C or O. Other letterforms need to be conceptualized differently to fit the representations. Above, the final twenty-six, with parameters. At left, initial typeform sketches.



in the folding and unfolding of the pie pieces, like folding paper fans. This idea was inspired by animated logo work by John Maeda for Dai Nippon Printing. What I enjoy most about this piece is catching the letters in mid-transition and playing with the emergent forms.

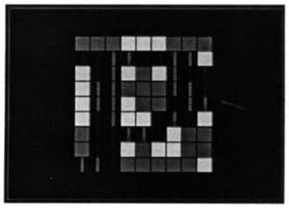
My one regret in part A is the design of the M. The M is problematic for several reasons, including its size extending far beyond the bounding square, its lack of symmetry, and its difference in style from the rest of the letters. When I showed this piece to type designer Jonathan Hoefler, he suggested the M could be more like the W. I agreed with his comment at the time, but upon further thought, I realized why I had settled on the somewhat unsatisfactory design as it was. In designing these abstracted letters, I found it necessary to decide, since I couldn't have them all, which details of the letterforms were most important. For me, I needed the bottom middle point of the M to convince me of its M-ness, while in a W, the corresponding point I think is more expendable; more important is the presence of two V's. This exercise for me introduces the concept of the "essence" of a letter, as I had to communicate letterforms with such pared-down, limited means.

Type me not, B

As the user presses the keyboard in Part B, a voice articulates the sound of each letter, which appears in an animation, then accelerates off the screen. Part B of this piece comes from a class problem asking us to represent different phoneme sounds visually as the corresponding letters are pressed. My solution featured a parametric, parts-based type font. This type representation makes it possible to animate different attributes of the type, changing, for instance, the weight of a letter over time. Seven parameters were specified – scale, cap line, stroke width, crossbar height, top height, mid point or x-height, and beard line or descender line – though only the first four were used for the upper case A through Z. In this font, each letter is defined as a set of instructions for building the letter based on the parameters given, in a similar but much simpler manner as Knuth's Metafont. By defining a letter as an instruction set instead of a list of coordinates, the shapes can be joined and jointed correctly, with strokes meeting at the appropriate places. Strangely, I designed most of the alphabet intuitively, picturing how I think the letters should look over the range of different parameterized values, and not actually looking at the result. In the end, I did not edit many of the codes after actually seeing instances of the letterforms.

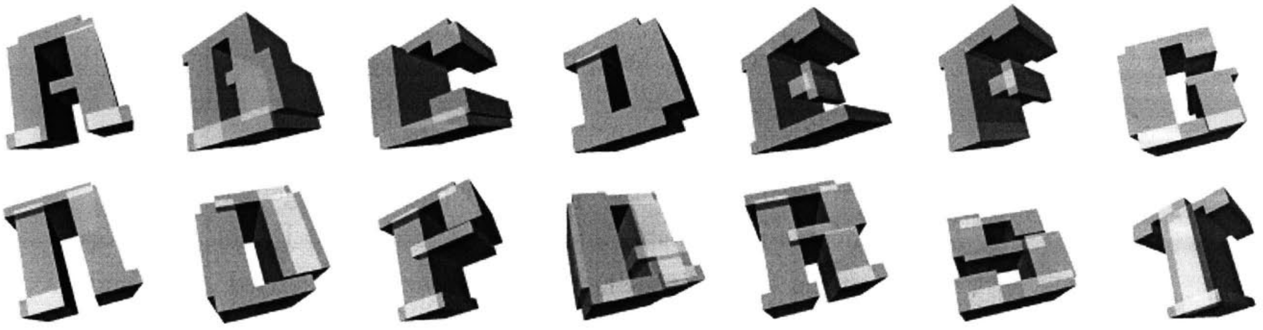
In part B, the animation of different letters corresponds with the kind of phoneme each letter presents. The explosive consonants – B, C, D, G, K, P, Q, T – "pop up," increasing size and stroke width during the transition in. The nasal consonants M and N, on the other hand, begin condensed and light, expand and darken to a bold regular width, then expand further to a light extended, all in time to a three-quarter second sound clip. In the Type me not piece, I circumvented the problem of having to compute and draw too many letters at once by having each letter slide off the screen after making its appearance. This device also made part B less about typing in words and more about the experience of hitting the keyboard and getting visual and audio feedback.

Representing a type font as a set of instructions rather than a predefined set of points seems like an attractive solution allowing for animating, parameterized typefaces, possibly in real-time. This is a somewhat unwieldy representation, however. What would be more appealing would be encoding a letter with some higher-level understanding of its shape and a typeface, higher-level knowledge of its component glyphs.



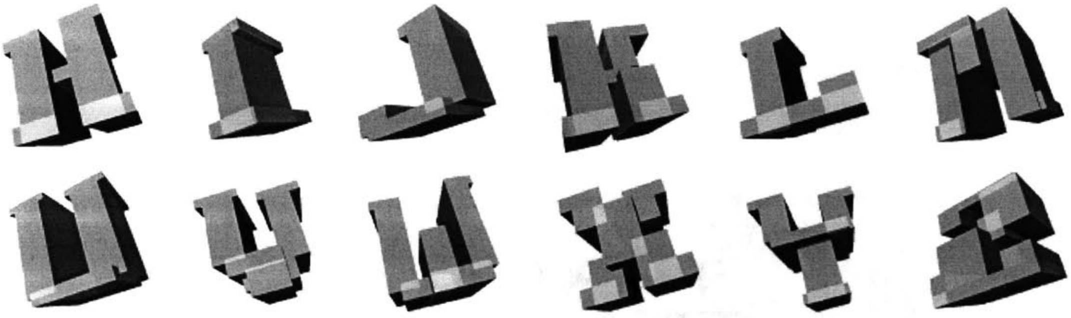
Type me not, C

In part C, letters appear in a nine by nine grid of tiles which are blue on back side and yellow on the front. When a letter is entered into the grid, each tile that comprises that letter flips over. If a letter H is typed into the grid at the starting, all-blue state, a yellow H



appears. If H is pressed again, the yellow H disappears again to blue. But if W is typed instead onto the H, some of the yellow tiles of the H turn back over to blue, and the grid is left with a seemingly random placement of yellow tiles. This design experiment is based on the computer graphics concept of "x-or": instead of painting a bit ON regardless of its previous state, you paint it ON if it the previous state was OFF or OFF if it was previously ON. In this design experiment, I was interested in the "intersections" of letters, the emergent patterns of superimposed letterforms. I do not think this experiment is especially successful at examining this idea of letterform intersections, primarily because of how the letters are designed within the grid. The letters are not entirely consistent in terms of x-height and the rasterization of curves. Also the letters which should descend lift up instead to fit within the square grid. I was disappointed because the emergent patterns seem like random collections of tiles rather than intelligent forms.

Part C does suggest one interesting concept, the storage of a stream of data within a small static space. In this scheme, the current state of the grid represents the entire succession of letters that have been fed into it. Within limitations, the set of letters which make up any pattern of tiles could be reconstructed, though without the order. Bill Verplank of Interval Research suggested the grid tiles could have range of states between ON and OFF, and perhaps a letter could make more of an impact into the grid if the key is held down for a length of time.



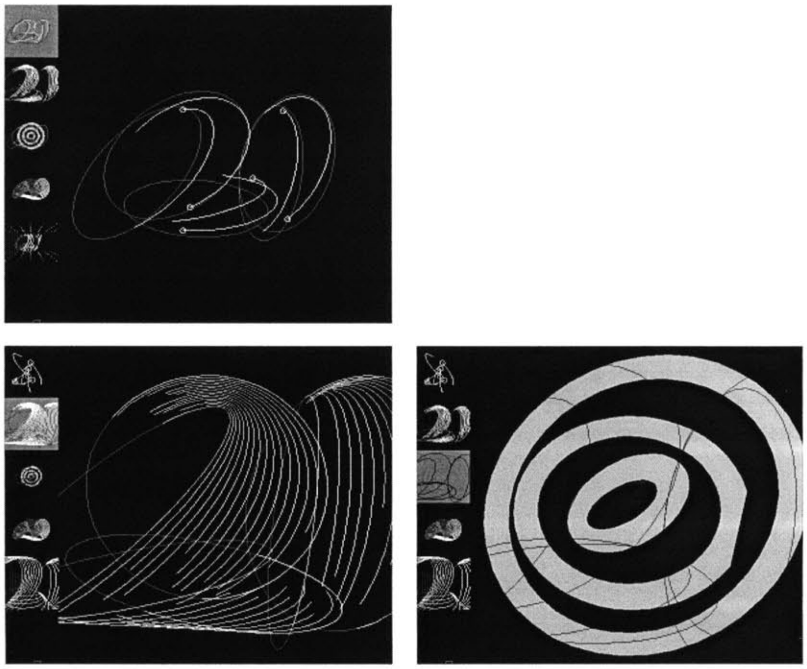
Left, and facing page, three-dimensional alphabet from Typeractive.

3.3 Typeractive

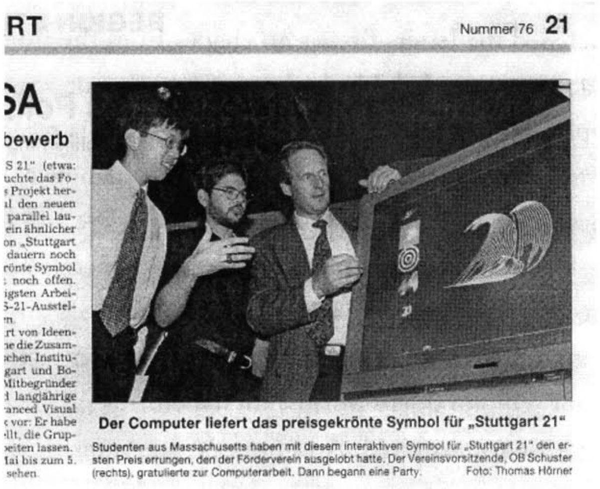
In the fall of 1997, I made my first attempt at a three-dimensional typeface with a design experiment called Typeractive. Similar in interaction to Type me not A, in this piece, the user types at the keyboard, and a letterform in the foreground transitions from one letter shape to another while swinging slowly from side to side to show the dimensionality. As the letters are typed in, small bitmap type also flies in from offscreen, and accumulates in a line of text in the background. The letterforms in this piece are composed of three to ten cube primitives which are scaled and translated, confined to a grid in three-space. Because the letterforms are modeled with the same set of parts, one letter can "morph" into another seamlessly.

In the Typeractive type font, each letter remains within the confines of a bounding cube and most touch all six sides of the cube. I sketched the letters first as 2D shapes from the front, looking for places an extruded, blocky letter could "break apart" into different z-planes: for example, the crossbar of the A is recessed. Most successful are the minimal moves in letters such as F, K, Q. In the end, these were not drastic moves – the shapes are still legible as letters.

Developing two-dimensional letter shapes as sculptural forms presents an interesting problem and requires a different kind of thinking from traditional type design. In contrast with Igarashi's elegant sculptural letterforms, the typeforms of this piece are quite crude and uninspired. Additionally, the transitions do not occur as



Three modes from the Stuttgart 21 animated logo piece. From top to right: Story, Architecture, Focus.



RT Nummer 76 21

SA

bewerb

S 21" (etwa:
uchte das Fo-
s Projekt her-
il den neuen
parallel lau-
ein ähnlicher
on „Stuttgart
dauern noch
rönte Symbol
: noch offen.
igsten Arbei-
S-21-Ausstel-
m.
rt von Ideen-
ne die Zusam-
chen Institu-
gart und Bo-
Mitbegründer
d langjährige
anced Visual
c vor: Er habe
llt, die Grup-
beiten lassen.
Mai bis zum 5.
sehen.

Der Computer liefert das preisgekrönte Symbol für „Stuttgart 21"

Studenten aus Massachusetts haben mit diesem interaktiven Symbol für „Stuttgart 21" den ersten Preis errungen, den der Förderverein ausgelobt hatte. Der Vereinsvorsitzende, OB Schuster (rechts), gratulierte zur Computerarbeit. Dann begann eine Party. Foto: Thomas Hörner

The author and colleague Rich DeVaul discuss the winning logomark with the mayor of Stuttgart.

charmingly as in the Type me not piece. Since block number one from letter G always moves to block number one from H and the ordering was unintentional, some of the inter-letter transitions seem awkward. Also, when the next letter has more or fewer components than the current, the extra blocks distract the eye by appearing from and disappearing into the lower-left back corner.

Designing in three dimensions affords many more possibilities than designing in two. The struggle is to create a design which works both as a truly three-dimensional form and, from a number of viewpoints, a two-dimensional design as well. And as a whole different issue, designing in virtual 3D requires more sophisticated, innovative creation tools beyond the software interface conventions we are accustomed to using.

3.4 Stuttgart 21

In March of 1998, our research group entered a competition to design a "living, breathing" symbol for Stuttgart 21, a project to revitalize the German city's transportation system for the twenty-first century. We received first prize in the competition with an interactive logo implemented in Java and a companion booklet. While this was a group project outside of the scope of my thesis research, I include it in this discussion because the design issues raised are relevant to this thesis research, and I was highly involved in the conceptual and implementation stages of developing the interactive logo.

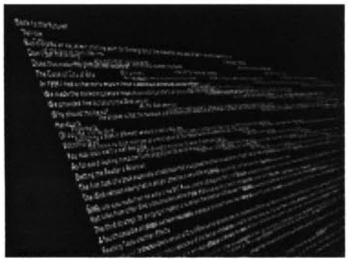
During the brainstorming stage of the project, we decided to create an animated and interactive symbol. I pushed early on for limiting the vocabulary of shapes to circular arcs in three-space. The resulting design uses six partial arcs to shape the numbers 2 and 1. In the interactive piece, the viewer can choose five different modes to see the logo. The first mode plays out as a looped animation describing the conception of the logo. The animation begins with an explosion from the center of the space, or a "big bang." The six arc pieces flow outward from the center into paths of electrons about a nucleus, then slowly transition into the orbits of planets about the sun. Finally, the "planets" disappear and the arcs move into place in the 21 logo shape. The remaining four modes explore aspects of space in two and three dimensions, allowing the viewer to interact with the symbol, rotate it, and move through it.

As in the Typeractive type design, the challenge is to develop a symbol that works both as a 3D shape and as a two-dimensional design. A further complication, and one we circumvented, was how to convey a dynamic, interactive logo in print media as a complete identity system. Prof. Maeda's reaction to the Stuttgart 21 design (besides general excitement about winning) was the sentiment, If there were more work like this out there in the world today, we would be better able to assess the quality of this design. This point is well taken: dynamic, interactive and three-dimensional design is only emerging as a medium, so the criteria for judging what works and what doesn't are not set. Currently, we have to rely on our intuition and the small body of work in this field. In the meantime, it seems audiences will be entranced by anything that reacts and moves in even a remotely interesting way.

3.5 Nutexts

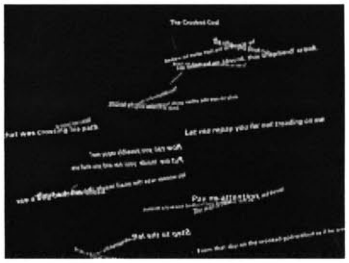
This set of experiments uses type to address the sculptural and architectural aspects of the screen space. In these design experiments, titled Nutexts and conducted during the summer of 1998, a body of text is positioned into a three-dimensional typographic composition. In these experiments, the focus is not so much on reading but on exploring implied 3D space. To this end, I used four different small- to medium-length texts: the introduction to William Mitchell's *City of Bits*, a short story by Amy Bloom, a poem by Samuel Keyser, and a web page from the MIT Media Lab site.

City of Bits



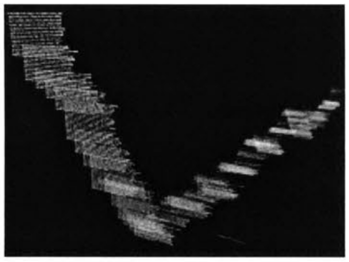
In the first Nutext experiment, each paragraph of the introductory essay of Mitchell's book *City of Bits* becomes a line extending into space. I used a simple device to move the text into three-space: the text moves back in space when it reaches a period, comma, colon, or semicolon. This simple choice changes a flat page of type into a texture, a pattern of rivers. The dimensionality becomes most noticeable when the text is moved, either translated or rotated in space. In general, these typographic designs look best in motion because in static form, the flat bitmapped type, while placed in an implied three-space, does not show dimensionality.

The Crooked God



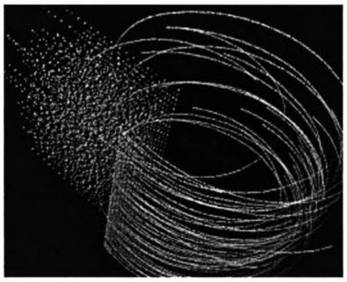
A poem by Samuel Keyser entitled "The Crooked God" in this Nutext experiment becomes a rotating sign post of straight and bent text lines. Each new line of the poem moves down and swings around an imaginary axis, so that as the entire type structure is rotated, the lines come to face the viewer in succession. Since this allegorical poem describes a god whose back was forever crooked, I play a bit with the type, placing a bend point between the quotation and the speaker in the lines of text featuring dialogue. This creates a bit of pause in reading because the quotation rotates to face the viewer before the rest of the sentence: "Pay no attention," swings around first, then, "said the snail."

Love is Not a Pie



In this design, Amy Bloom's short story "Love is Not a Pie" is displayed in space in a programmatic way to see if the structure of the narrative is evident in the structure of the text itself. At the end of each paragraph, the insertion point moves back a fixed amount in z and up a small fixed amount from the last line, so that the next paragraph position depends on the length in lines of the previous paragraph. When viewed from the side, the structure of the story in terms of the lengths of its paragraphs becomes visible. The shape of the story falls at a fairly constant and steep rate for the first third of the text, then remains somewhat level with a few stops and jumps for the rest of the story. Upon further examination, the reader notices the beginning of the story consists of a first-person narrative description of the character's mother, while the rest of the story comprises flashbacks to the character's childhood with dialogue, hence the short paragraphs. This design experiment suggests that we can visualize and learn something about a text, or any data, just by examining a simple metric, such as the size of component elements.

Media Lab web page



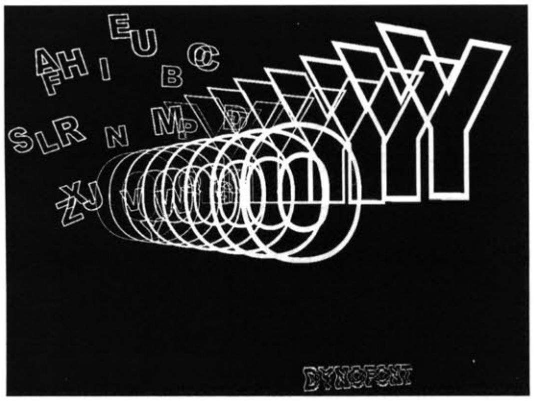
In the fourth experiment, a page from the MIT Media Lab web site becomes an abstract spiraling structure, with each paragraph of the text page becoming a curving strand, anchored by a cloud of letters composed of the names of Media Lab members. This design experiment came from Prof. Maeda's suggestion that I try to combine two contrasting ideas together. While the least functional of the four examples as a readable text, this design is the most visually striking and represents the best case I can make for typography as sculpture.

3.6 Letterspace

In fall of 1998, I developed an interactive piece based on interletter transitions in a TrueType font. In this piece, the user holds a 3D magnetic field sensor in each hand, each sensor representing a letter on the screen. When the user moves the two sensors, the gestures in space are mapped to different letters, and the letters on screen gently transition to new letters, demonstrating a coupling between the fluid visual transitions and the smooth physical motions.

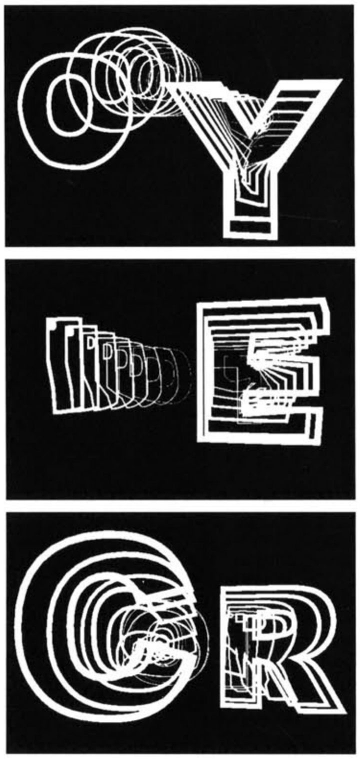
To be able to draw the interpolations between letters, I use the FreeType package to pull out the A through Z contour data points

At left, the author performs the letterspace system. As the user moves the two hand sensors, the gestures are mapped to letter changes on the screen.



Letterspace onscreen image.

At right, three letter transitions in motion. As the front letter changes, the transition propogates backward, creating a trail of letterforms.

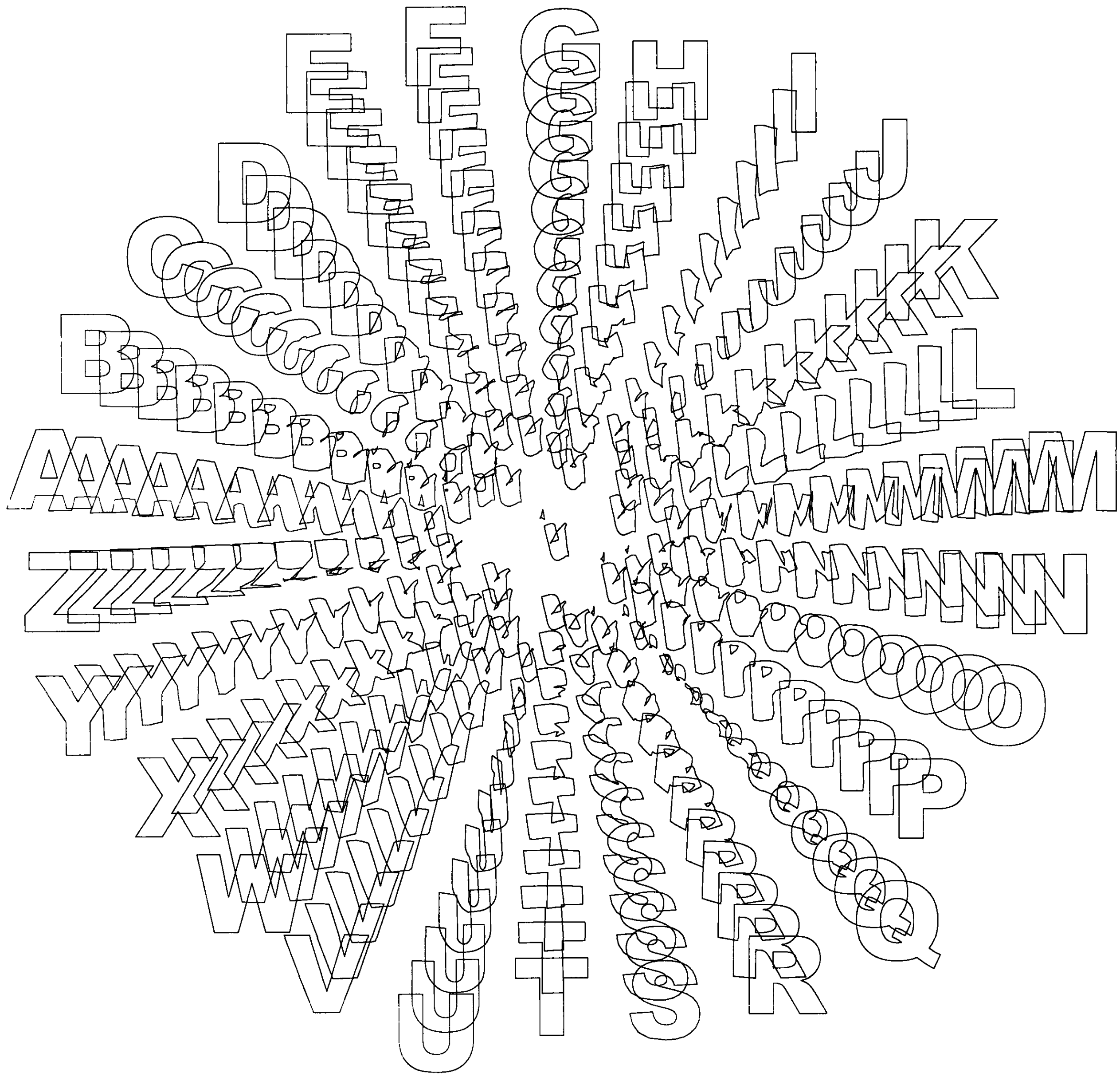


from a Truetype font. I then sample each outline at a number of points along the contour to create a polygonal representation which can approximate the curved segments along with the straight. Since I use the same fixed number of points for each letter, it becomes possible to transition from one letter shape to another, simply by interpolating the points. Because letterforms have differing numbers of contours when they are drawn in this "outline" mode (most have one, some letters have two contours, B has three), the extra contours need to come from somewhere; in Letterspace, they appear from and disappear into a fixed point in another contour.

In a second mode in Letterspace, triggered by pressing the spacebar, the space rotates in response to the relative position of the two sensors. In addition, a trail of letters appears behind each letter on screen, echoing the motions of the letter. As a letter transition occurs, the letter propagates along this trail to the back. At certain points in time, this trail becomes a three-dimensional extruded shape with one letter at the front and a different at the back.

Letterspace represents my first design experiment which considers the performance aspect of this piece. Users of this piece become engaged for a number of reasons: often they see a connection between the physical actions and the visual response, but not an immediate correlation, so they become engaged in "figuring it out"; along more basic lines, they are also engaged physically – they have to stand and flail their arms about. In one case, I saw a fellow graduate student actually twisting his wrists and even his whole body in breakdance-like moves in response to the twisting letter transitions onscreen.

While I consider this piece generally successful, I find a few aspects problematic. One is how the letter transitions occur. If I were to revisit this piece, I would experiment with different kinds of transitional animations besides the basic linear interpolation I use here. Instead of moving each point along a straight vector, I would

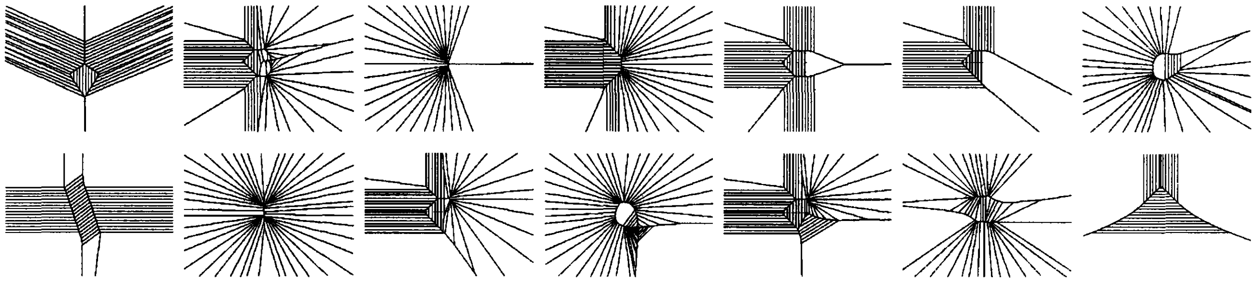


After developing the mechanism for discretized font outline paths, I also spent some time exploring transitioning letterforms in print by creating scripts to output Adobe Illustrator Postscript code. I created a few images, including the graphic above. In this image, the 26 letters feed into the "averaged" character in the center.

consider using a curved Bezier path, motivated by the shapes of the contours. Another aspect of the Letterspace piece which does not quite work is the swinging globe of letters in the upper-left of the screen, which serves as a key to the gestural mappings to letters. Many users have difficulty seeing this legend as a three-dimensional sphere or understanding that it rotates back and forth just to make the dimensionality more obvious. Finally, the gesture recognition could be smarter, classifying more than simple relative directional vectors.

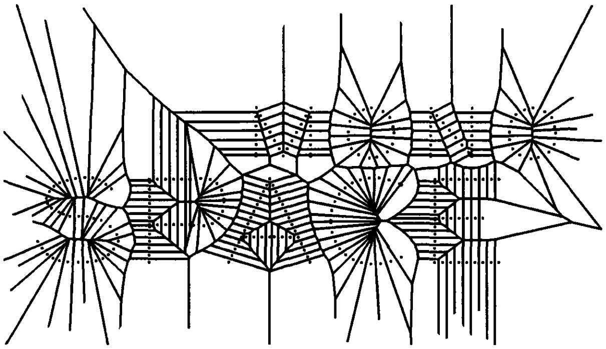
3.7 Voronoi type

During January of this year, I developed a small design experiment which used Voronoi diagrams to explore how type can affect the space in which it lives. In a Voronoi mathematical diagram, source points are placed in a space. The algorithm then constructs the borders to build cells. Each cell in the diagram surrounds a source point, such that all positions within a given cell are closer to its source than any other point in the diagram. In this experiment, I drew dots at regular intervals along simple archetypal line-drawn letterforms then fed those dots into a Voronoi diagram. In this way, we can see the impact a letterform has in "breaking up" a space. Curved letters cause radial wedges to form, while straight-lined letters create narrow rectangles. The Voronoi diagram, in a sense,

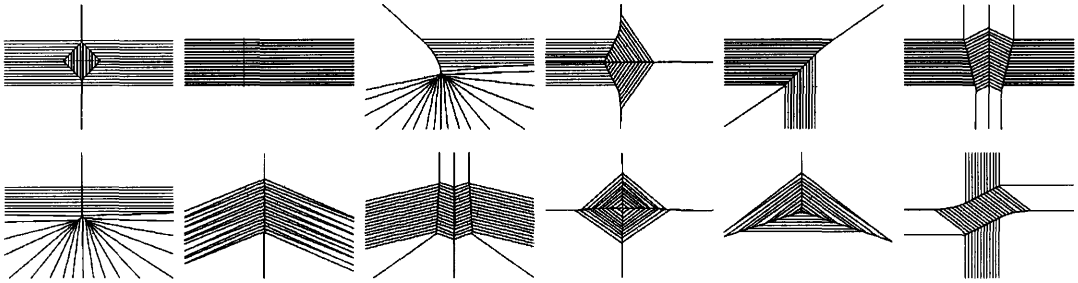


Above and facing page, basic wireframe letters are placed into Voronoi diagrams, resulting in the patterns above. Top row, A through M; bottom, N through Z.

Right, the word 'monospace,' placed into a Voronoi diagram.

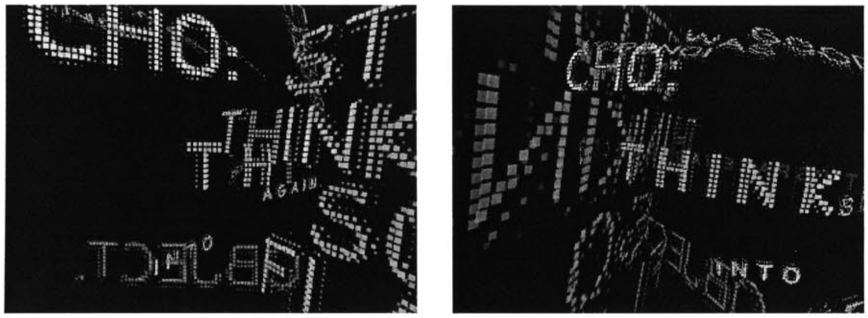


tends to "augment" the shapes of the letters. The diagram for the letter K, for example, is broken into three regions corresponding to the three strokes of the letterform. I also experimented with placing a word into a Voronoi diagram. In the MONOSPACE image, we can see the borders which the algorithm constructs between letters, for example, the irregular border between the letter C and E in SPACE.



This technique points to a possible algorithmic method for studying letterspacing and perhaps computing kerning between letters automatically.

When I tried experimenting with type in Voronoi diagrams, I intended to create type which gets "blown" into a system of cells, causing the cells to push away and adjust. I imagined an interactive piece in which the user inflates a letterform within a mass of bubbles. This belongs to a set of research concepts dealing with physically- and mathematically-based particle systems, featuring at-



Frames from the Message animation. Left, frame 2310. Right, 2945.

oms which attract, repel, clump, disperse, systems in which each individual cell knows how to move to affect the whole, rather than the overall motion being scripted. These are ideas I would like to return to in the future.

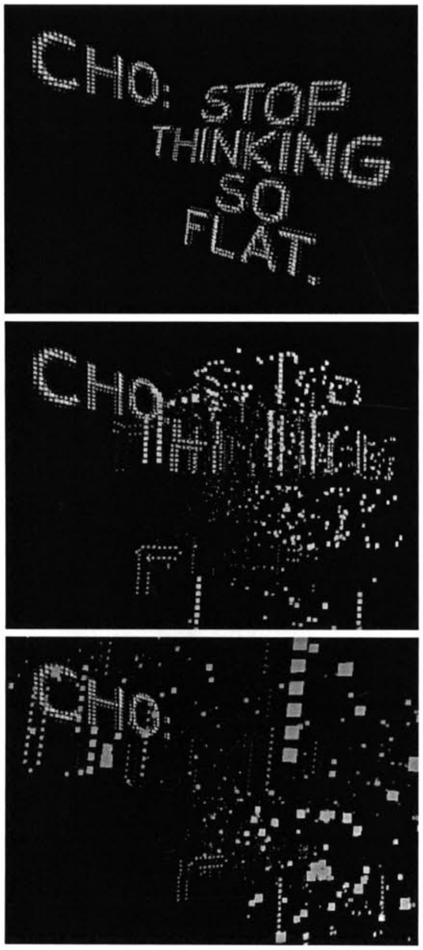
3.8 Message: Stop Thinking So Flat

Prof. Maeda sent me a status message in February, telling me in what direction he would like me to go with my research. His message read:

> Cho: stop thinking so flat. try to move into space again. voronoy [sic] was good experiment. do not focus on 'center' object. think 'environment'/ 'space'/architecture.

In response to his suggestions, I built a typographic animation in three-space, using his message. Instead of using a bitmapped font for the text, I use a type font designed from scratch, featuring a metaball-inspired computational model. In this type face, a letterform is represented as a skeleton connecting 2D "anchor" points with straight and curved segments. This skeletal letter exerts a force from its strokes which pushes up against a two-dimensional grid. This grid in turn can be visualized in different ways including a collection of floating square tiles or a connected bumpy surface. Additionally, the font can change the "amplitude" of its bumpiness and the sampling resolution of the grid. This representation results in a type face which looks conventional from the front, but when viewed from the sides in 3D space, also shows a bit of dimensionality in z. This becomes useful because flat bitmapped type has the problem of disappearing when it is rotated 90 degrees to the side – if a typeface is dimensional, it can be used to give depth or position cues in the virtual three-dimensional screen space. This type model is described in more detail in the appendix section.

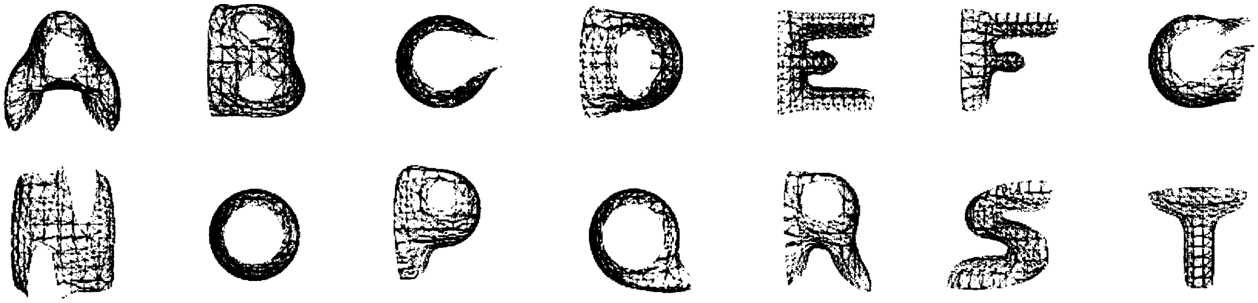
To create the type animation, I built tools which allow for setting key frames at different times for the type and also the camera. These key frames affect several properties of the type, including scale, translation, rotation, color, grid sampling, and grid amplitude.



At right, the words "Stop thinking so flat" decompose into tiles which fly toward the reader. For a moment, the type deconstructs to pure image.

The camera's key framing allows for setting the position and rotation. Also, the key framing tools allow for setting the type of interpolation from one point to another: linear, ease in, ease out, or ease in and out. In the Message animation, each sentence occurs in a different "stage" in the space. After each sentence appears, the camera moves to a new point in space, but the text remains, building on to what came before to create, in the end, a freeform typographic structure.

This piece is effective for its dramatic use of virtual three-dimensional space and its coupling between message and visual display.



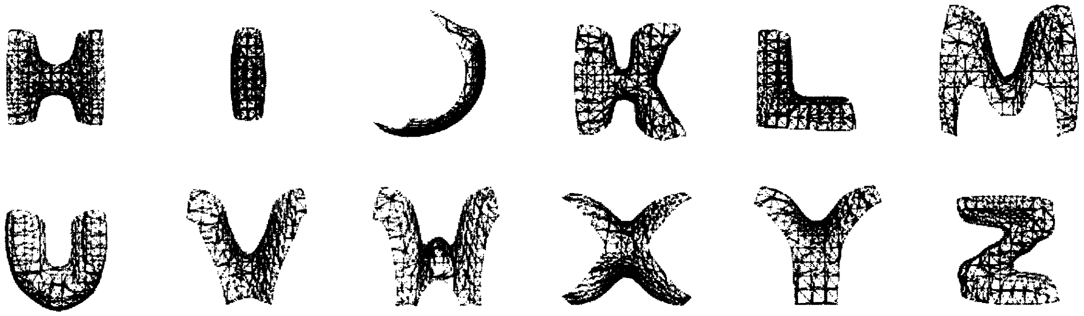
Above and facing, typeface from Alphabet Zoo, based on implicit surfaces

Perhaps the most interesting moment of the animation occurs during the phrase, "Stop thinking so flat." After appearing on screen, the words fly outward toward the viewer, abstracting into a mass of square tiles floating through space. For this brief, surprising moment, the type becomes illegible as words and deconstructs to pure image. This typographic animation, along with the growing field of motion graphics, suggests where we might be headed as communication design combines with cinematic storytelling to present new ways of conveying textual information.

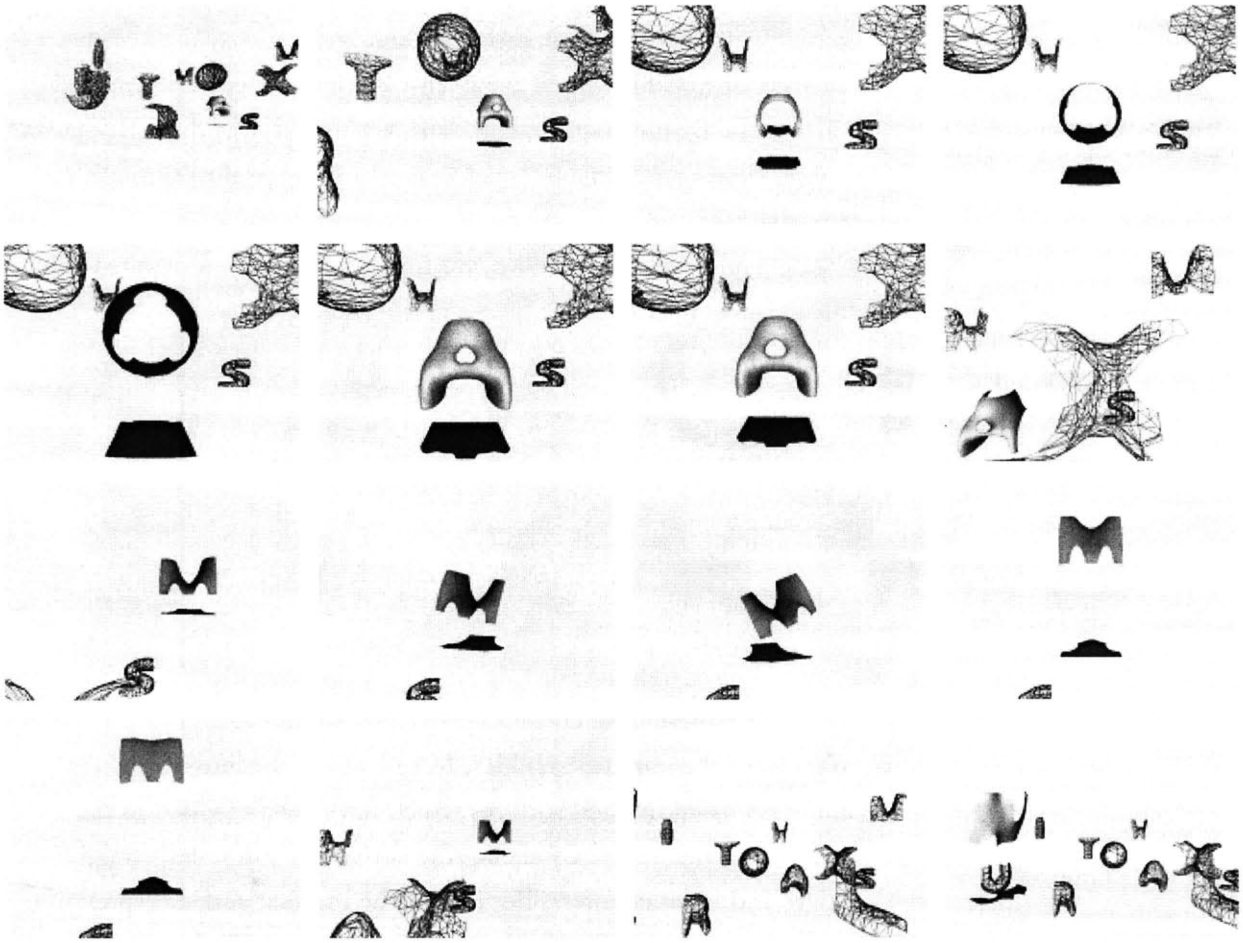
3.9 Alphabet zoo

In the next design experiment I became interested in designing organic forms. To this end, I began exploring metaballs and implicit surfaces. While much computer-based 3D geometric modeling is done with basic primitives such as lines, boxes, and spheres, metaballs construct surfaces which are contours, or isosurfaces, through a scalar field in 3D. In this design experiment, I build letterforms using implicit surfaces, then create a virtual three-dimensional world in which the twenty-six letters become animated characters. By pressing the keyboard, the user moves through the space, viewing the organic typeforms which come alive through expressive motions.

Beyond beveled extrusions, we find few spatially-motivated solutions for representing conventionally two-dimensional type in three-dimensional computer graphics. In this piece, the letterforms are modeled using implicit surfaces which have been cropped in the three coordinate axes. By slicing the surfaces, I can reveal straight edges and letter counters. Because of the implicit surface representation, the letters take on fluid, shell-like forms which draw cues from the human body. Most successful are letters such as B, M, N, and W, which depart from a two-dimensional shape to a true sculptural form. These letters push and pull on our expectations of the two-dimensional shapes.



After designing the forms, I create animations for the letters to make them “dance” in space, using the letterforms themselves to suggest certain kinds of motions. The M, with its grounded legs and



In these sixteen frames from the
Alphabet Zoo in action,
the viewer visits a hungry letter A
and the playful M.

sweeping form, for example, “twitches” near the ground plane, then leaps into the air. Its neighbor N, however, is not so successful; he leaps, only to fall flat on his face. These looping motions give personalities to the typeforms, turning static forms into living characters.

While I wanted to modify the underlying metaball representation to perform these animations, I could not do so in real-time. Instead, I transform the coordinate space in subtle ways, playing with interpolations between the normal letter representation and a cylindrical projection in the x, y, or z direction. These transforms give a fluidity to the moving forms which would not be possible with animations of only translation, scale, and rotation.

When the user presses a letter on the keyboard, the camera moves smoothly from the current position to the viewpoint for the new letter. With this interaction, the viewer travels from letter to letter, visiting the twenty-six exhibits in the Alphabet Zoo.

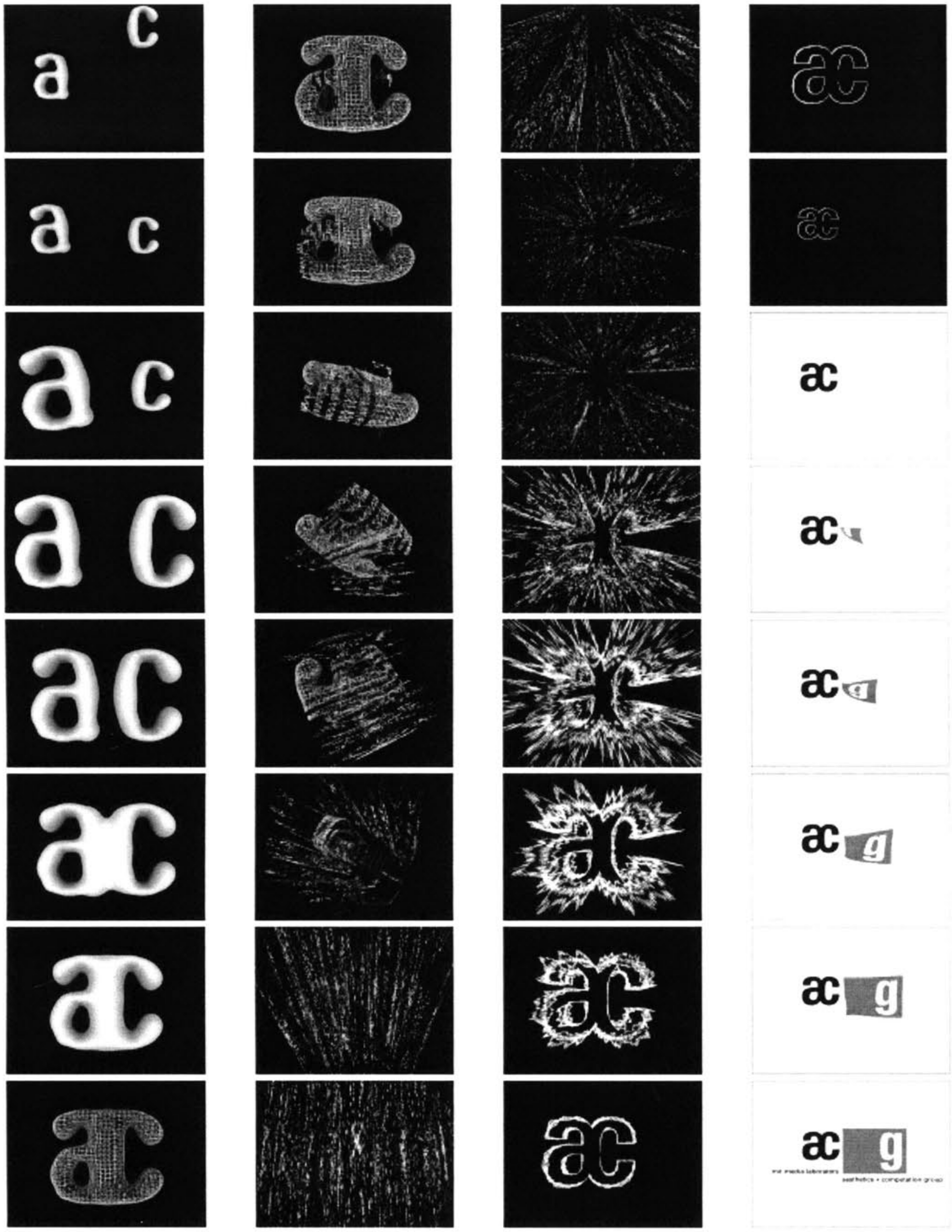
This piece represents a living illustrated storybook in which the letters of the alphabet take on colorful personalities. This design experiment points toward how the playful and educational aspects of a children’s book can combine with carefully crafted and sophisticated three-dimensional graphics.

3.10 ACG logo movie

In March of 1999, I was asked to create a short animation to use as an introduction to the Aesthetics and Computation Group. I began from the ACG logomark. Since the lowercase A and C are joined into a custom ligature in this logomark, I decided to animate the fusing of these two letters, using the implicit surface technique I explored in the Alphabet Zoo project. I wanted the blobby surfaces somehow to transition into the actual Univers Bold Condensed of the logo, however, so I used a series of transformations in three-dimensional space to create the 2D symbol.

In the animation, the A and C begin as characters, falling from the sky, then bouncing toward the viewer. They pause, then join into a combined AC shape. As the letters join, the blobby surface flickers, revealing the wireframe structure beneath. Next the wireframe flies apart as the form tilts and rotates and the component lines move into the background to create an abstract pattern. The lines fragments rotate into a radial pattern, then stream in toward the center of the frame, revealing the outline shape of the logomark's AC ligature. The AC moves into place on the left-hand side of image, then flashes from white on a black background to black on white. Finally, the red G field flows fluidly from the AC, and the text lines "mit media laboratory aesthetics + computation group" fade into view.

This piece combines many of the technical elements of the previous design experiments into a single, rapid-fire, eight-second animation. Part of the problem with such a short piece with so many transitions, however, is that the transforms "fly right by" on the first viewing. Though the ACG logo movie is not interactive, per se, the animation does compel the viewer to want to watch it more than once. This piece suggests an issue which arises also in the design of special effects action sequences for film: to excite while still preserving continuity and comprehensibility.


mit media laboratory
aesthetics + computation group

Thirty-two frames from the eight-second ACG logo movie

4 Analysis

4.1 Mechanics: Bits & pieces

Over the course of this thesis research, I have built up a set of technical nuts and bolts to use in digital works. Some of these techniques involve basic concepts of 2D and 3D graphics, others are specific algorithms which originate in other uses and applications. Following is a chart indicating the technical elements used in each experiment.

	Design experiment		*Technical elements*
3.1	***Pliant type***	*Letter dance*	bezier, one-off lag
		LMNOP	bezier, linear interpolation
		Sleepy, Aim, Oh	linear interpolation, move away from or toward point
3.2	***Type me, Type me not***	*A*	ease-in interpolation
		B	parameterized form
		C	x-or mode
3.3	***Typeractive***		interpolation, two- to three-dimensional space
3.4	***Stuttgart 21***		two- to three-dimensional space, key framed animation
3.5	***Nutexts***	*City of Bits*	translated space
		The Crooked God	rotated space
		Love is Not a Pie	translated space
		Media Lab web page	rotated space
3.6	***Letterspace***		bezier, linear interpolation, TrueType conversion, pattern matching, rotated space
3.7	***Voronoi type***		bezier, voronoi diagram
3.8	***Message: Stop Thinking So Flat***		key framed animation, parameterized form, translated and rotated space, metaballs
3.9	***Alphabet zoo***		implicit surfaces + metaballs, key framed animation, translated and rotated space, coordinate transformation
3.10	***ACG logo movie***		bezier, move away from or toward point, implicit surfaces + metaballs, key framed animation, translated and rotated space, Postscript conversion

These technical elements compound upon one another throughout the thesis research. Some of these bits and pieces require a bit of explanation.

Beziers

A bezier is a type of parameterized curve, named after a French engineer who used them for the body design of the Renault car. In general, a bezier is specified by two end points and any number of control points between them which "pull" on the curved line. In this thesis research, I used a special case of the bezier curve with four points. The equation for this special case is given by

$$B(t) = P_0(1-t)^3 + 3P_1u(1-t)^2 + 3P_2t^2(1-t) + P_3t^3$$

where P_0 and P_3 are the end points and P_1 and P_2 are the middle control points; t varies from 0 to 1.

Interpolation

Interpolation is a method for animation in which a parameter varies from one value to another over time. In linear interpolation, this variation occurs along a straight line. Ease-in and ease-out is accomplished within linear interpolation by stepping along a straight line at a non-constant velocity, for example with x^2.

Move away from or toward point

In this simple technique for creating expressive motion, the control points of an outlined shape move away from or toward a specified source point, for example, the mouse cursor. The expressiveness and fluidity of motion comes about because control points are affected more or less based on proximity to the source point.

Parameterized form

One way to make a form which can change its shape is to make it parameterized. Parameterized form involves developing a template for an object and axes for setting values for changing the object. In a sense, parameterizing a form means giving the form more intelligence about itself. In this research, I have designed computational models for parameterized type and have experimented with animating the type by changing these parameters (i.e. weight, width, scale) over time.

X-or mode

In the normal paint mode, the background color has no effect on the paint operation. If you paint a black dot, it appears as a black dot, no matter on what color background it was painted. In the x-or paint mode, however, the background color affects the paint color. In the simplest terms, if you paint the same object twice while you are in x-or mode, the object disappears. When in x-or mode in a two-color system, a pixel reverses its state if it is painted upon.

Key framed animation

In animation on the computer, a key frame represents a marker describing the status of an image at a precise moment. The animator builds key frames to define the motion of an object, then the computer computes the intermediate frames between keys necessary for a smooth animation.

Two- to three-dimensional space

One interest of mine, which I have explored in this thesis research, is the interaction between two- and three-dimensional space. In general, this involves the interplay between reading an image as a flat surface perpendicular to the viewer and seeing an image as a collection of planes, lines, or forms in a perspectival space. In the context of 3D graphics, this can mean taking the xyz coordinates of objects in space, projecting them to the xy of the flat screen space, then manipulating the three- to two-dimensional conversion in some way.

Translated and rotated space

These are vague terms for the kinds of three-dimensional space I use in this thesis research. In translated space, the elements exist in such a way that the dimensionality is revealed by translating, or by moving the camera in a pan. In rotated space, the dimensionality is realized by rotating the space about an axis.

Pattern matching

Pattern matching, or pattern recognition, is a general research area that studies the operation and design of systems that recognize patterns in data. It encloses subdisciplines such as discriminant analysis, feature extraction, error estimation, cluster analysis, grammatical inference and parsing. Some of the areas to which pattern matching is applied are image analysis, character recognition, and speech analysis. In this thesis research, I use rudimentary pattern matching to recognize gestural inputs.

Voronoi diagrams

The Voronoi diagram of a collection of geometric objects is a partition of space into cells, each of which consists of the points closer to one particular object than to any others. These diagrams, their boundaries (also called medial axes) and their duals (Delaunay triangulations) have been reinvented, given different names, generalized, studied, and applied many times over in many different fields. Voronoi diagrams tend to be involved in situations where a space should be partitioned into "spheres of influence" and has applications in such diverse studies as protein molecule volume analysis, cell growth, terrain modelling, and animal territorial behavior.

Coordinate transformations

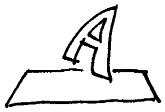
One way of manipulating space is to transform the coordinate system, by mapping the Cartesian xyz space to a cylindrical or spherical space. In this research, I use this transformation to "warp" the shape of three-dimensional objects.

Metaballs + Implicit surfaces

In contrast to traditional parametric surfaces, implicit surfaces can easily describe smooth and intricate shapes. An implicit surface is described by a closed equation. For example, the equation for a sphere is $F(x,y,z) = x^2 + y^2 + z^2 - r^2 = 0$. In a metaball representation, particles in space create a density field, where the density attributed to a particle decreases with distance from the particle location. The metaball surface is the isosurface which is formed by connecting the locations in the field having equal density.

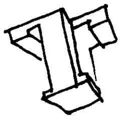
4.2 Categorization of experiments

In this section I categorize the ten design experiments according to three metrics, spatial qualities, interaction type, and linearity.

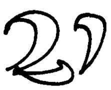


1. Pliant type

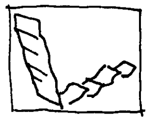
2. Type me not



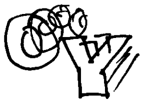
3. Typeractive



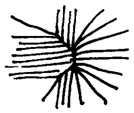
4. Stuttgart 21



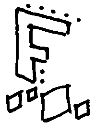
5. Nutexts



6. Letterspace



7. Voronoi type



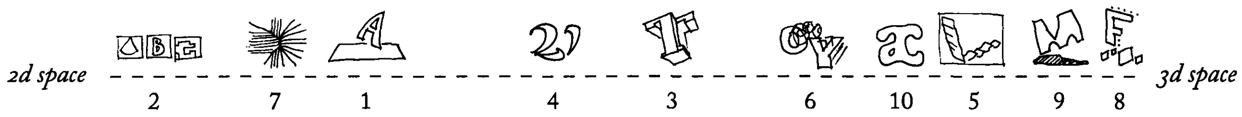
8. Message

9. Alphabet zoo

10. ACG *logo*

Kinds of space

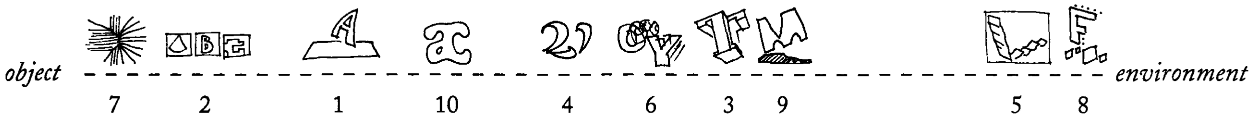
The design experiments from this thesis research span from treating the screen plane as a flat, two-dimensional design to thinking of the screen as an immersive, three-dimensional space.



The projects in general become more three-dimensional in feeling as time progresses. I began working in 2D because I felt more comfortable dealing with interactions of elements in a flat plane. At first I was only interested in using “2 and a half D,” meaning using a 2D computer environment while creating the illusion of dimensionality in the screen plane by changing the scale and brightness of elements. I was reluctant to use true 3D graphics at first because of my general disinterest in the computer graphics aesthetic, often embodied in the attempt to create a hyper-realistic shaded, textured world. The turning point for me was the Nutexts project, in which I explored a drastic use of space and discovered the visual richness of using a virtual 3D environment.

Object vs. environment

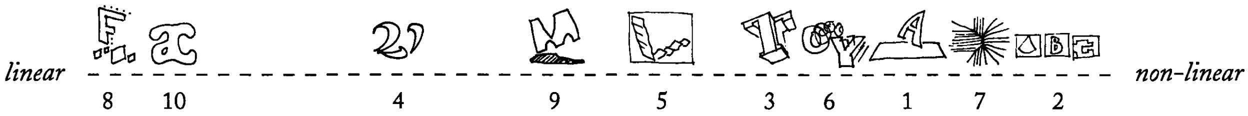
In these pieces, we see the difference between interacting with letterforms as prominent objects and moving through an environment populated by type.



Toward the left hand side, we see interactions which are akin to interacting with a toy, a "thing" to play with. Toward the right, we see projects more involved with creating an environment, a "place" to be, inhabit, and explore. The Alphabet Zoo project represents a combination of these two extremes; here the user interacts with single characters, who also exist in an abstract space.

Linearity

The research experiments also demonstrate the range between a linear, scripted story and a freeform, nonlinear environment with many entry and exit points.



This metric is related to the degree of user-interactivity: generally speaking, the more interactive, the less linear the piece becomes. As this thesis research progressed, I found myself more interested in creating experiences with a narrative aspect. Two of the later projects, the Message and ACG logo, are entirely linear, scripted animations.

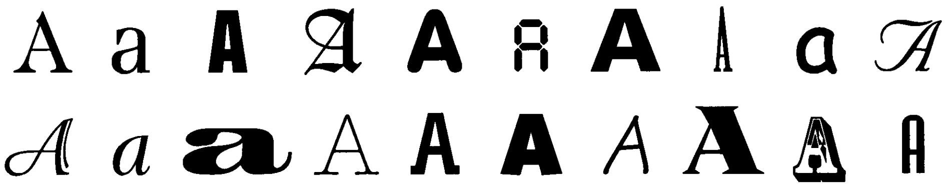
People who discuss digital media tend to focus on interactivity as the main difference between new media and traditional design. To me, interactivity is a bit of a red herring, especially when dealing with narrative. Many of the experiments into combining stories with interaction involve giving the audience simple choose-your-own-adventure-type options to decide the course of the plot. Most of these experiments are not successful, however, because viewers want to be told a story and immersed in an imagined world. Giving this kind of choice pulls the audience out of the cinematic experience. What interests me is using the medium of the computer to draw the viewer in. This may involve encouraging interaction with the computer, but the interaction is not necessarily what makes for an engrossing experience. Often an animation can be just as compelling as something interactive if I can see a beautiful, surprising, and clear design intention coming through.

5 Thoughts on type

5.1 Letters as symbols

Letterforms represent the building blocks of written communication. As such they have evolved in many human cultures. Our own Roman alphabet has its origins in ancient Phoenician writing from many centuries before the Christian era. The alphabet gained some letters and lost others over a period of centuries to form the set of 26 we use in the English language. From an early age, we learn to write and recognize the alphabet. Children's television programming encourages youngsters to see letters as dancing characters that speak out the sound they represent. We also learn early on that each letter has an uppercase and a lowercase, which most readers recognize subconsciously. Ingrained in our minds at such a young age, letters have a certain peculiarity as symbols of abstraction, which when combined to form words, gain a semantic connotation.

What is especially interesting about letterforms is the way in which we can recognize the letter A, for example, in such a vast array of images. This makes it difficult to determine the criteria for what makes a letter what it is. How would you describe all of the qualities of the letter A to someone who had never seen the Roman alphabet? You can imagine the scenario:

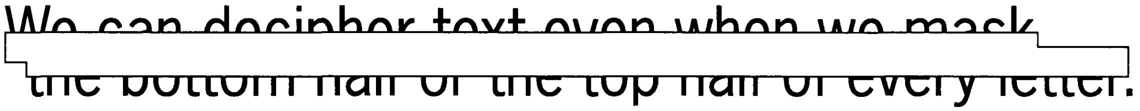


What is an A?

> An A has two straight diagonal lines meeting at a point at the top and a horizontal bar that touches the two sides somewhere in the middle. Except sometimes the two sides are more up and down, and then they meet at the top in a flat line or a curved cap. Sometimes an A is "lower case," meaning it's usually a straight vertical line on the right side with a curved line that

> starts at the top of the vertical line, bulges out to the left and down, then meets the vertical line again near the bottom. But sometimes the lower case A also has a hooking piece connected to the top of the right-hand-side vertical line. It sort of curves over like a hanging plant – oh, never mind.

Even if we pick out a criterion which should be foolproof – the capital letter A comes to a point at top – we can find a number of exceptions to the rule. Is it possible to pinpoint the "essence" of a letterform? We see through reading experiments that we can decipher text even when we mask the bottom half or the top half of every letter. If we know that a shape is a letter, we can puzzle out ambiguous letterforms by looking at the context of the word it is in and the other letters around it. This usually occurs at a subconscious level, as when we try to decipher someone's bad handwriting. We know how a prototypical alphabet looks, but we are also accustomed to so many different kinds of writing that it may not be possible to determine the "essence" of a specific letter beyond, "I know a 'G' when I see one." In *Metaphysical Themas*, Hofstadter suggests:

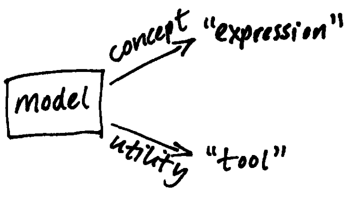

We can decipher text even when we mask
the bottom half or the top half of every letter.

> Clearly there is much more going on in typefaces than meets the eye – literally. The shape of a letterform is a surface manifestation of deep mental abstractions. It is determined by conceptual considerations and balances that no finite set of merely geometric knobs could capture. Underneath or behind each instance of 'A' there lurks a concept, a Platonic entity, a spirit. This Platonic entity is not an elegant shape such as the Univers 'A', not a template with a finite number of knobs, not a topological or group-theoretical invariant in some mathematical heaven, but a mental abstraction – a different sort of beast. Each instance of the 'A' spirit reveals something new about the spirit without ever exhausting it. (p. 279)

In some ways a letterform is like a human face – an infinite range of possibilities within a given structure, two eyes, a nose, a mouth. And when training a computer how to see a person's face in an image, we can have reasonable success rates of recognition – reasonable, but never perfect. We can see the same result when we try to teach a computer to recognize type. The best commercial optical character recognition software package claims "over 99% accuracy," which sounds good until you read the disclaimer: "on laser quality documents using standard fonts."

A big difference lies between training a computer to recognize a range of type fonts and teaching the computer to generate the same diversity of typefaces. Computer scientists who try to develop a "universal font," an uber-font computational representation which, given parameters, can spit out any number of typefaces, quickly run up against a brick wall. Even the Metafont project, with its versatile set of parameters, represents only a tiny fraction of the kinds of typefaces that exist today. In my opinion, this kind of research into computational type is a dead end, which is why I have focused in this thesis research on the expressive aspects made possible through custom computational models.

5.2 On computational models



Two paths for developing computational models

The difference between engineering design process and graphic design process can be seen in the early stages of project implementation. Developing computational models for visual phenomena is often difficult because the engineer is more concerned with functionality than with the end result. Software engineers are trained to think about the "right way" of solving a problem. Usually this involves encoding levels of data abstraction on top of the machine's internal code. In theory, this means that the more successive layers are added, the more readable and understandable the code becomes. Part of the problem, however, is that the more layers, the further you get from what actually happens on the screen. The design process in the ACG is different: since we can see the problem from both the design and engineering viewpoints, we can work at the problem from both ends and plan the visual outcome accordingly. Working in this manner, we can certainly be surprised and swayed by the medium along the way, but the initial visual concept is present before we ever start to code.

This thesis research involves developing custom computational representations for type which make certain specific types of manipulation possible. Inherent in this work, however, is a trade-off: these models allow for new kinds of manipulation, but they make others difficult, if not impossible. Most of these computational models, for example, do not allow for an entire class of image filters: blurring, sharpening, pixelizing.

In general, a computational model is used to represent a system in terms a computer can understand. This model can represent natural phenomena like the motion of waves or the interactions of people in an office space, or it can represent abstract concepts such as the parsing of spoken language or the structure of a database. When building a computational representation for something that exists in another form, we are faced with the problem of formalizing the phenomenon in a systematic way. The model needs to be broad enough to allow for the complexities and particular eccentricities of the system, but should also have a handle on the important higher-level patterns which can make the model especially useful. On the computer, we generally represent type as Postscript or TrueType outlines. Type design, however, is truly a high-level system which can be perceived in many layers: from the design of specific characters and components such as bowls and serifs, to the overall visual weight and axis of a character set, to more general levels, encompassing different type styles, different faces, and the specifics of written language in general. With this in mind, we have to wonder whether the outlines offered by Postscript and TrueType representations make for a sufficient computational model.

At right, image from an early research project, an exploration of gestural sketch input into a three-dimensional space. At the last minute, I infected an abstract study with type to make it more accessible.

In this research, I have only touched on these concepts and the design possibilities of custom models for type. One aspect of computational typography which I did not get a chance to pursue was to give "intelligence" to a type representation. A type representation could be more effective in a functional sense (i.e. type could know how to adjust letterspacing automatically) but also could lead towards new kinds of expressive typography. What if a letter could "know" about itself, its composition of straight and curved members, its image density, its phoneme sound, its frequency in the English language, its neighbors? What if a word could be more intelligent, too, knowing its function in a sentence, and a sentence knowing its function in a paragraph? These questions suggest a new kind of active, conscious type which could affect the act of reading in interesting ways. These intriguing questions will have to remain unanswered for the time being.

5.3 Type can be a crutch

Type represents visual language. We look at type, and we know what it is about. Letters are symbols everyone recognizes; when the symbols appear in sequence, they form words that people can understand as higher level concepts. Typography is used to convey many kinds of information, from stock quotes and weather reports to poetic personal experiences. Type also has an intensely graphic quality, with contrasting shapes and forms. When type is placed on a page, a strong foreground-background relationship emerges. Type can be used to define spatial relationships or layered to create texture and type as image. We can appreciate type for its many wonderful attributes, but if we depend on it to make an artistic exploration "happen," we must realize that we have made a compromise. Type can be a crutch.

Many of the artists of the twentieth century explored type in the context of visual media such as painting and collage. In his artworks, Jasper Johns used symbols such as flags, maps, and color charts as pictorial elements because he wanted to use symbols we all recognize so that any changes he made to them became clear. We find that in developing creative works in the computational medium, we want to rely on type and other easily-accessible symbols instead of exploring pure form, color, line and motion. I have certainly found this in my design research at the Media Lab. In spite of an advisor who urged me away from type exploration, I always came back, in fact devoting a whole thesis to the subject.

I do not mean to say that type is somehow easy. Type designers devote their entire lives to creating beautiful, effective typefaces; good typographic design is also a challenge. What I mean to say is that if we decide to use type in our exploratory work, we have a compulsion to make the type really good – not merely present, adding its two or three cents worth to a piece, but embodying the qualities of good typography: legible, readable, meaningful, pristine, intense, intelligent, heartfelt, inevitable. A tall order, I know, but we have to try.

6 Concluding remarks

6.1 A design philosophy

In the emerging field of new media, we see a tendency to make design obscure or unpredictable. This trend hides the underlying content in layers in an effort to bring an aspect of mystery or sophistication. This path is easy and seductive, misusing the computer's tendency to conceal the message, or lack thereof, by making the work oblique or unreadable. New media designer Gigi Biederman of Post Tool Design states:

> People are always criticizing interactivity because it isn't really interactive, it is just selection-driven, choice, choice, choice. Our work has some of the same problems, but we have included elements of randomness. When you select something, you don't really know what you are selecting and you don't know what you'll get, so you respond to the unexpected.
> (Karam, p. 28)

The use of randomness in particular can entrap the digital designer. On the computer, you can have absolute creative control, but this wide open space of possibilities can be intimidating. If you let the computer use its random number generator to spit out values, you can often create interesting visual phenomena quickly but may ultimately end up cheapening the work. The random function can give you an instant color palette or pattern of dots, but it's not one you have chosen as a designer. Random noise can be a useful tool, but it is a tool that needs to be handled carefully. In his book *Design By Numbers*, Maeda suggests:

> If computation can be thought of as a compact means to render complex images, then computation using random numbers is perhaps the single most powerful method to express complex graphics compactly. Unfortunately, the outcome is difficult to predict, which seems to be the whole point of using random numbers. There are many better ways to become completely lost in systems of noise [than the ones presented here], and if this is the path you wish to pursue, then I hope that you get lost in earnest. (Maeda, 1999, p. 249)

Using randomness to direct a user's experience, as Biederman suggests, is even more of a cop out. Even if the goal is (already a discomforting thought) to confuse the user deliberately, using computer-generated randomness is a cheap way to do so. If the viewer complains about not being able to understand the piece, the designer might suggest that the user's frustration was intended. More likely, however, the viewer just gives up and picks up a book or goes to watch television.

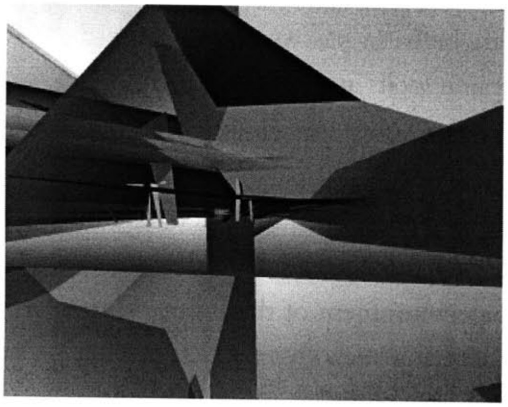
In three years in the ACG, I have designed dozens of interactive pieces in courses and independent research. One aspect that runs as a common thread through these works is the desire to make things understandable, while still surprising and delightful. Computation can often seem cold or sterile as a medium, especially in the realm of fractal patterns or virtual reality experiences. One of my goals has been to create works that push away from this cyber-world, to give a "human" quality to computational design. How can we do this? On the surface level, this means giving a designer's eye and touch to the work, being attuned to the details of line, color, type, motion, and interaction. On a deeper level, bringing humanity to digital design can involve drawing inspiration from the beauty of nature and from life. It also can mean bringing something personal, one's own humor and personality, to the work. It is my goal to bring something honest and heartfelt to this medium.

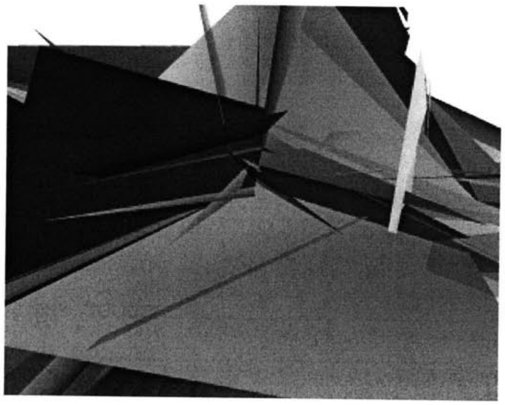
6.2 What is this all for?

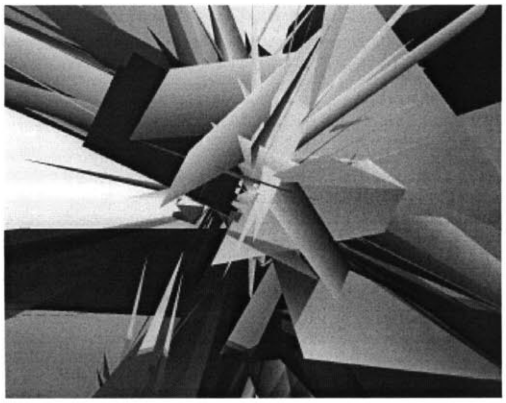
The big question, a few answers. This thesis research contributes to a study of the creative manipulations of typography made possible through custom computational models. I use an exploration of the parameters involved in type design, what Hofstadter calls "twiddling knobs," to bring out expressive uses of type. In a sense, this research introduces new design "knobs" and expands the space of possibilities afforded through homegrown digital representations of type. These typographic possibilities can have applications in static, dynamic, and interactive media.

Another contribution of this thesis research, suggested especially by the Message animation, involves the future of communication design in the context of dynamic typography and the computer. The rise of animated type in film, broadcast, and the web has affected the way we see and decode text, combining the act of reading with cinematic storytelling. The computational medium allows us to push and pull on the boundary between passive and active reading experiences. This thesis research looks at ways we can explore this boundary and think about approaching the computer display as a dynamic, living reading environment.

Finally, a confession. For my first two years in the ACG, I found myself hesitant about my research at the Media Lab and unsure about why I was doing it. In the environment of the Lab, and MIT at large, I had a difficult time justifying such abstract design research. It is in the past year that I have come to realize why this research is relevant today. John Maeda has remarked that the reason print graphic design is at the stage it is today is that it relies on a vocabulary and tradition from the visual arts, especially from 20th century studies of abstraction. And the reason interactive media design is in the state it is today is that it does not have a correlating fine art tradition to draw from; instead, digital media has to extrapolate, synthesize, and adapt from what works in other media. This thesis research explores the region where computation, design, and art merge, while avoiding the constraints and limitations of any particular utilitarian or industrial need. I am reluctant to call our studies into the computational medium art, but that is where we are headed. We are exploring this region to discover what lives at the crossroads.



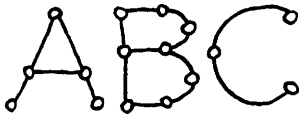




Accidental images resulting from Alphabet Zoo code gone awry

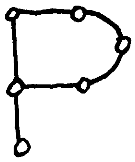
A Implementation notes

In this appendix section, I describe the type representation I used for the Message: Stop Thinking So Flat animation in more detail.



The type model used in the Message is based on a skeletal representation. Each letter is specified as a list of "center" points and a list of "links" which connect the center points. The links can be of two types, straight lines through two points and curved arcs specified by three points.

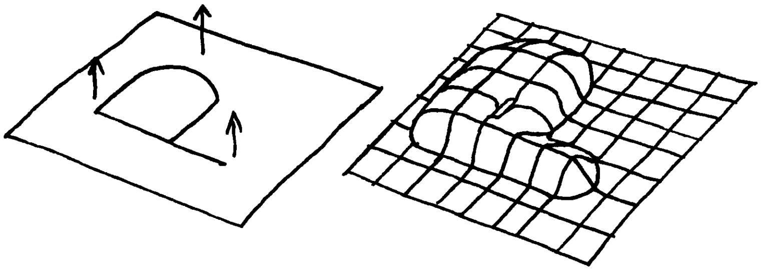
This data is stored in a pcFrame. For example, following is the description of a capital P:



```
P = new pcFrame();
x1 = en*1.05;
P->addCenter(0.0, cap);   // top left
P->addCenter(x1*0.55, cap); // top right
y1 = cap*0.45;
P->addCenter(x1*0.9, (y1+cap)*0.5); // top bulge tip
P->addCenter(0.0, y1); // mid left
P->addCenter(x1*0.55, y1); // mid right
P->addCenter(0.0, 0.0); // bottom left
P->addLink(LINE, 0, 1, -1);
P->addLink(LINE, 3, 4, -1);
P->addLink(ARC, 0, 3, 5);
P->addLink(ARC, 1, 2, 4);
width = x1*0.9;
```

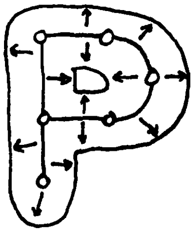
This representation is parameterized. In the code above, "cap" stores the value for the cap-height, and "en" stores the en width. These values can be changed globally to affect the entire font of characters.

In the Message animation, the letter from a pcFrame next is placed in a grid, where it exerts an influence by pressing up against the 2D plane:



The pcGrid uses an exponential “bell curve” to determine the how the letter pushes against the grid. At a particular x, y point, the z scale in a grid is given by

```
scale=exp(-pow((x-cX)/cS,2)/2)*exp(-pow((y-cY)/cS,2)/2)*cA;
```



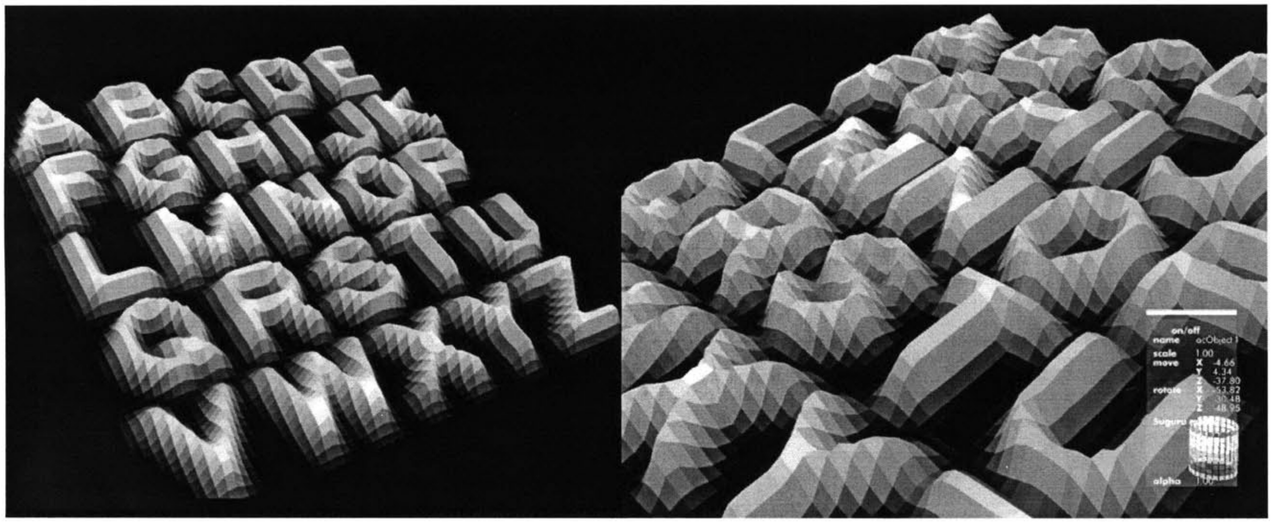
where cA is the amplitude of the bell curve, and cS is the standard deviation, or the “spread” of the curve.

The pcGrid draws letters by stepping along the link paths and setting values at each grid point. Following is the code used for calculating the grid values.

```
void pcGrid::setGridValues() {
  Center* ctrs;
  int i, j, a, b;
  pcFrame* fm;
  Link* lnks;
  Vec2 scale, trans;

  // clear out grid
  for (i=0; i<gridX; i++) {
    for (j=0; j<gridY; j++) {
      putValue(i, j, 0.);
    }
  }

  // handle links between frame centers
  float x1, y1, x2, y2, x3, y3, ls;
  Center c1, c2, c3;

  for (a=0; a<numFms; a++) {
    fm = &fms[a];
    scale = fm->scale;
    trans = fm->trans;
    if (fm->getNumLinks() > 0) {
      ctrs = fm->getCenters();
      lnks = fm->getLinks();
      for (b=0; b<fm->getNumLinks(); b++) {
        if (lnks[b].type == LINE) {
          // straight line
          // traverse path from lnks[b].c1 to lnks[b].c2
          // assigning increased grid values
          c1 = ctrs[lnks[b].c1];
          c2 = ctrs[lnks[b].c2];
          x1 = c1.x * scale.x + trans.x;
          y1 = c1.y * scale.y + trans.y;
          x2 = c2.x * scale.x + trans.x;
          y2 = c2.y * scale.y + trans.y;
          ls = fm->getLS();
```

```
     drawLine(x1, y1, x2, y2, c1.s, c2.s,
          c1.a, c2.a, ls);
    } else if (lnks[b].type == ARC) {
     // bezier arc
     c1 = ctrs[lnks[b].c1];
     c2 = ctrs[lnks[b].c2];
     c3 = ctrs[lnks[b].c3];
     x1 = c1.x * scale.x + trans.x;
     y1 = c1.y * scale.y + trans.y;
     x2 = c2.x * scale.x + trans.x;
     y2 = c2.y * scale.y + trans.y;
     x3 = c3.x * scale.x + trans.x;
     y3 = c3.y * scale.y + trans.y;
     ls = fm->getLS();
     drawArc(x1, y1, x2, y2, x3, y3,
         c1.s, c2.s, c3.s, c1.a, c2.a, c3.a, ls);
    }
   }
  }
 }
}
```

In the Message animation, I visualized the grid using perpendicular square tiles. In earlier experiments, I also tried using a continuous surface of tiles. After experimenting with different visualizations of the type, I returned to the square tile visualization in the animation because it is less computationally intensive.


on/off
name acObject 1
scale 1.00
move X -4.66
Y 4.34
Z -37.80
rotate X -53.82
Y 30.48
Z 48.95
Suguru
alpha 1.00

References

Carter, Matthew. "Typography and Current Technologies." Design Quarterly 148, Walker Art Center and MIT, 1990.

Cho, Peter. "Pliant Type: Development and Temporal Manipulation of Expressive, Malleable Typography." Bachelors thesis, Massachusetts Institute of Technology, 1997.

Hofstadter, Douglas. Metamagical Themas: Questing for the Essence of Mind and Pattern. BasicBooks, New York. 1985.

Gershenfeld, Neil. When Things Start to Think. Henry Holt & Company. 1999.

Igarashi, Takenobu. Igarashi Sculptures. Robundo Publishing Inc., Tokyo. 1992.

Ionesco, Eugene., Massin, and Cohen. The Bald Soprano. Grove Press. 1956.

Ishizaki, Suguru. "Typographic Performance: Continuous Design Solutions as Emergent Behaviors of Active Agents." Doctoral thesis, Massachusetts Institute of Technology, 1996.

Karam, Kendall. Playing with Code: Interview with Gigi Biederman and David Karam. AIGA Journal of Graphic Design. Vol. 16, No. 2, 1998.

Kinross, Robin. Modern Typography: an Essay in Critical History. Hyphen Press, London. 1992.

Knuth, Donald. Computer Modern Typefaces. Addison Wesley, Reading, Massachusetts. 1986.

Kram, Reed. "System Models for Performance." Masters thesis, Massachusetts Institute of Technology. 1998.

Lewis, Jason. "Dynamic Poetry: Introductory Remarks to a Digital Medium." Masters thesis, Royal College of Art, 1996.

Licko, Zuzana., Poyner, VanderLans, and Wild. Emigre. Rosbeek, The Netherlands. 1998.

Lupton, Ellen and Miller, J. Abbott. Design Writing Research: Writing on Graphic Design. Kiosk, New York. 1996.

Maeda, John. Design by Numbers. MIT Press, Cambridge, Massachusetts. 1999.

Maeda, John. Flying Letters. Digitalogue, Tokyo. 1996.

Miller, J. Abbott. Dimensional Typography. Kiosk. 1996.

Mills, C. and Weldon, L. "Reading Text from Computer Screens." ACM Computing Surveys. Vol. 19 No. 4. 1987. pp. 329–358.

Potter, M. "Rapid Serial Visual Presentation (RSVP): A Method for Studying Language Processing." New Methods in Reading Comprehension Research. Erlbaum. 1984. pp. 91–118.

Shamir, Ariel and Rappoport, Ari. "Feature-Based Design of Fonts Using Constraints." International Conference on Raster Imaging and Digital Typography. 1998.

Small, David. "Expressive Typography: High Quality Dynamic and Responsive Typography in the Electronic Environment." Masters thesis, Massachusetts Institute of Technology, 1987.

Spencer, Herbert. Pioneers of Modern Typography. Lund Humphries, London. 1969.

Strassmann, Steve. "Hairy Brushes." ACM 20, no. 4. August 1986. pp 225-232.

Wong, Yin Yin. "Temporal Typography: Characterization of Time-Varying Typographic Forms." Masters thesis, Massachusetts Institute of Technology, 1995.

3739-[illegible]