Expanding Information Access through Data-Driven Design

Danielle Bragg

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2018

Reading Committee:

Richard Ladner, Chair

Alan Borning

Katharina Reinecke

Program Authorized to Offer Degree:
Computer Science & Engineering

University of Washington

Abstract

Expanding Information Access
through Data-Driven Design

Danielle Bragg

Chair of the Supervisory Committee:
Professor Richard Ladner
Computer Science & Engineering

Computer scientists have made progress on many problems in information access: curating large datasets, developing machine learning and computer vision, building extensive networks, and designing powerful interfaces and graphics. However, we sometimes fail to fully leverage these modern techniques, especially when building systems inclusive of people with disabilities (who total a billion worldwide [168], and nearly one in five in the U.S. [26]). For example, visual graphics and small text may exclude people with visual impairments, and text-based resources like search engines and text editors may not fully support people using unwritten sign languages. In this dissertation, I argue that if we are willing to break with traditional modes of information access, we can leverage modern computing and design techniques from computer graphics, crowdsourcing, topic modeling, and participatory design to greatly improve and enrich access.

This dissertation demonstrates this potential for expanded access through four systems I built as a Ph.D. student: 1) Smartfonts (Chapter 3), scripts that reimagine the alphabet's appearance to improve legibility and enrich text displays, leveraging modern screens' ability to render letterforms not easily hand-drawn, 2) Livefonts (Chapter 4), a subset of Smartfonts that use animation to further differentiate letterforms, leveraging modern screens' animation capability to redefine limitations on letterform design, 3) Animated Si5s, the first animated character system prototype for American Sign Language (ASL) (Chapter 5), which uses animation to increase text resem-

blance to live signing and in turn improve understandability without training, and 4) ASL-Search, an ASL dictionary trained on crowdsourced data collected from volunteer ASL students through ASL-Flash, a novel crowdsourcing platform and educational tool (Chapters 6-7). These systems employ quantitative methods, using large-scale data collected through existing and novel crowdsourcing platforms, to explore design spaces and solve data scarcity problems. They also use human-centered approaches to better understand and address usability problems.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

ALPHABET: a set of basic units or letters, often representing linguistic components or sounds, used to represent a language in text.

AMERICAN SIGN LANGUAGE (ASL): a visual-spatial movement-based language, which is the primary language of the Deaf community in the United States and parts of Canada.

BLINDNESS: a type of visual impairment characterized by complete or near-complete vision loss.

CHARACTER SYSTEM: a set of symbols with rules for using them to represent a spoken or sign language in text.

DEAF: a cultural identity (with a capital D), or a hearing status (with a lowercase d).

LATIN: the letterforms (the traditional A-Z) that form the basis of English and other Latin languages (e.g., Spanish, French, and Italian).

LETTERFORM: the shape (or more generally, the visual or tactile design) of a letter of an alphabet.

LOW VISION: a type of visual impairment, characterized by acuity not near 20/20 even with best correction (e.g., with glasses).

VISUAL IMPAIRMENT: decreased vision that impacts daily life and persists even with best correction (e.g., with glasses).

ACKNOWLEDGMENTS

I am deeply grateful for my advisor Richard Ladner, who introduced me to accessibility and has been an incredibly supportive mentor, PhD advisor, and collaborator. Throughout my PhD work, he has been a consistent, grounded source of guidance. This work would not exist without him. I am also grateful for the open academic environment of UW's CSE department where I completed this work, and in particular to Lindsay Michimoto for her support in having me join. A big thank you to my committee for their thoughtful feedback as well: Alan Borning, Katharina Reinecke, Adam Kalai, Meredith Ringel Morris, and Jaime Snyder.

I would like to thank Adam Kalai for being such a creative, generous, and supportive mentor, and for hiring me to work on Smartfonts, and later Livefonts, as an intern. His initial idea to create alternate English scripts developed into and inspired a significant portion of this dissertation. Our work on Smartfonts and Livefonts would not have been possible without our other fantastic collaborators, Shiri Azenkot, Kevin Larson, and Ann Bessemans, who helped shape the projects with practical guidance, deep knowledge, and experience. I am also thankful to Microsoft Research New England for supporting much of the work on Smartfonts and Livefonts through internships, and for being such an inspiring place to work.

I would like to thank Raja Kushalnagar for his support and collaboration on Animated Si5s, and the students at Gallaudet University for their participation in our workshop. I also thank the Harlan Hahn Endowment Fund for supporting the project.

I would like to thank Kyle Rector for her collaboration and guidance on the development of our ASL-Search dictionary, and for welcoming me as a young graduate student joining the project. I thank SigningSavvy for granting permission to use their sign videos in the development and deployment of ASL-Search and ASL-Flash, and for helping to publicize our work. I thank Erika

Teasley for her diligent work compiling signs from textbooks, matching those signs to videos, and inputting feature evaluations, and Tisha Trongtham for helping to develop the dictionary front-end. I also thank the National Science Foundation and Google for supporting our work on the dictionary.

I am very grateful to UW's ASL teachers, Lance Forshay, Kristi Winter, Michael Cooper, for their generosity in discussing, providing feedback, and generally supporting my work. Lance Forshay and Kristi Winter also taught me basic ASL and about Deaf culture. They have greatly enriched my life and my work.

My supportive peers have made all the difference in completing this work, in particular Shiri Azenkot, Catie Baker, Gagan Bansal, Cynthia Bennett, Caitlin Bonnar, Anna Cavendar, Jim Chen, Eunsol Choi, Felicia Cordeiro, Alex Fiannaca, Or Gadish, Eunice Jun, Naga Katta, Chris Lin, Lauren Milne, Iris Odstrcil, Kyle Rector, Arvind Satyanarayan, Jessica Tran, and Vicky Yao. Additionally, a very special thank you to László M. Lovász for always being there to encourage and support me.

I would like to thank the professors who have helped me along the way by encouraging me to do research, and giving me opportunities to do so: Hyeong-Ah Choi, who encouraged me to apply for graduate school and has been an incredibly generous and supportive mentor; Miriah Meyer, who guided me through my undergraduate thesis with practical advice and kindness; Hanspeter Pfister, who supervised my undergraduate thesis, and has been nothing but supportive; Leana Golubchik, who introduced me to research through an NSF REU program at USC; and Michael Brenner, who encouraged me to apply to that NSF REU program and made it possible to do so.

Finally, I cannot thank my family enough: my parents for the incredible eduction they gave me; my mother for her unconditional love and support; and my brother, friend, and colleague Jonathan Bragg for his unending love and support.

DEDICATION

To my mom.

Chapter 1
INTRODUCTION

This dissertation focuses on improving access to information for low-vision readers and sign language users. Text-based resources are pervasive, but present accessibility problems to these groups. Low vision can make reading difficult, in particular on small personal device screens; and sign languages lack a standard written form, rendering many text-based resources, including books and search engines, incompatible with sign languages. At the same time, sophisticated data-driven design techniques spanning computer graphics, crowdsourcing, topic modeling, and participatory design have been developed that could be leveraged to improve access. In this dissertation, I show that leveraging these technological affordances can greatly improve access to information if we are willing to break with traditional text-based resource design.

1.0.1 Motivation

Text-based resources are pervasive in modern society. Books, newspapers, journals, and other written publications are fundamental to education, work, civic engagement, and pleasure, and are available both in print and digitally. The internet – estimated to comprise over 1.5 billion websites[1] – is largely text-based, from website content to search engines used for indexing and navigation. Interpersonal communications have also become increasingly text-based as email, text/SMS, social media posts, and instant messengers have infiltrated our professional and personal lives. Text-based displays are also fundamental to everyday communication – for example, street signs, billboards, restaurant menus, and instruction manuals all typically involve text. Text-based content generation is supported by word processors, text editors, and email clients for countless businesses, schools, organizations, and individuals. Though much of society and the technical world communicates

[1] `http://www.internetlivestats.com/total-number-of-websites/`

through written text, text-based platforms present accessibility problems to millions of people.

In particular, text can be inaccessible to people with poor vision. Visual impairment can be defined as decreased vision that impacts daily life and persists even with best correction (e.g., glasses or contact lenses). Blindness refers to complete or near-complete vision loss, while low vision refers to acuity not near 20/20 with best correction. Worldwide, there are about 285 million people with visual impairments, the vast majority of whom (about 246 million) have low vision [115]. In addition, nearly everyone's vision declines with age, resulting in presbyopia, the inevitable and irreversible decrease in the eyes' ability to focus [110, 2, 43]. The emergence of personal computing devices can compound legibility problems. To fit content on a small screen, text must be small. This small text size can make computing devices unusable to people with visual impairments, and decrease usability for sighted users straining to read small letters, especially without glasses at hand. Using extreme magnification, a common solution on larger screens, is intractable on small devices because too little content can be displayed at once (perhaps only one letter at a time).

Text-based resources can also be inaccessible to sign language users. Sign languages are movement-based languages without a standard written form. Worldwide, there are about 70 million deaf people using a sign language as their primary language [113]. Many hearing people use sign languages, too. For example, ASL has become the third-most popular language in U.S. higher education, behind only French and Spanish, and has increasing enrollment [108]. The absence of a commonly accepted written form prevents sign language users from reading and writing in their language of choice. It also makes it difficult to search for sign language content, as search engines are designed for written languages, and consequently require text queries. Translating text into sign language videos or animated avatars is not always possible, and fails to provide the functionality of text, which represents concepts in the abstract, provides visual context and flexibility during reading, and enables fast, inconspicuous content creation and editing. Expecting people to use English or another written language is also inappropriate, as English is an entirely different language, English literacy rates are low[2], and sign language is often preferred.

---

[2]as the average deaf high-school graduate reads at a fourth-grade level [67]

Even though technology has exacerbated these accessibility problems by proliferating inaccessible text-based systems, technology also provides opportunities to redesign resources to improve accessibility. Worldwide, our letterforms have historically been constrained to what is easily producible by hand – monochromatic lines, curves, and dots. Our mediums for displaying text have advanced well beyond traditional pen and paper, removing these constraints on letterform design. In particular, today's screens easily support rich displays involving colors, complex shapes, and even animation that can be incorporated into letterform design. Furthermore, the emergence of customizable personal devices allows for individual adoption of novel scripts without language reform or mass adoption for the first time in history. Modern topic modeling and data collection techniques also allow for the development of robust, scalable sign language search techniques not based on text.

In this dissertation, I present systems that leverage modern computing capabilities to improve access to information for low-vision readers and sign language users. I explore how we can redesign English letterforms for modern screens to improve legibility for low-vision readers. These scripts expand typography to include letterforms defined by color or animation.[3] I also explore incorporating animation in an ASL character system to create text that more closely resembles the movement-based language and potentially improves learnability. Finally, I explore using topic modeling over crowdsourced data to create a feature-based ASL dictionary robust to query variability and sign corpus size.

1.0.2 Thesis Statement

This dissertation provides work in support of the following thesis statement:

> *Information access can be greatly expanded, in particular for low-vision readers and sign language users, by redesigning character systems and sign language search through data-driven techniques involving computer graphics, crowdsourcing, topic modeling, and participatory design.*

[3] Many figures in this dissertation itself should appear animated when viewed with Adobe Reader.

1.0.3 Contributions

In support of this thesis statement, I designed, implemented, and evaluated a set of information access systems in support of low-vision English readers and ASL users: Smartfonts, Livefonts, an animated ASL character system prototype, ASL-Search, and ASL-Flash. To build effective solutions, I employed methodologies that incorporate direct and indirect input from target populations: crowdsourcing through existing and novel platforms, optimization over collected data, topic modeling, and participatory design. This section highlights the main contributions of each project. In presenting these projects, I use "we," as these projects are all the result of joint work with collaborators.

Chapter 3 Our guiding research questions are: *Can we improve text legibility by leveraging computer graphics to redesign our letterforms? How can we design these alternate scripts in a principled way?* In response, we present Smartfonts, scripts that re-design letterforms, leveraging computer graphics (e.g., color, shape, and spacing) to improve or enhance the reading experience. Our key contributions are: 1) the idea of smartfonts, which expand typography beyond monochromatic lines, curves, and dots, to which letterforms have historically been constrained worldwide, 2) a demonstration of how a smartfont design can be optimized with respect to crowdsourced perceptual data to improve particular aspects of the reading experience, 3) a methodology for evaluating script legibility without training participants to read the script, and 4) initial evaluations of smartfont legibility and learnability. This work on Smartfonts supports the thesis statement by demonstrating that legibility barriers to text-based information can be addressed by leveraging computer graphics and crowdsourced perceptual data to redesign letterforms. Smartfonts were initially published in [22].

Chapter 4 Our guiding research questions are: *Can we further improve text legibility by leveraging animation capabilities of computer graphics to differentiate letterforms? How can we design these animated scripts in a principled way?* In answer, we present Livefonts, a subcategory of Smartfonts that uses animation to differentiate letterforms, making text newly legible at approximately half the size of traditional letterforms. Our key contributions are: 1) the introduction of

animation as a defining feature of letterform design, 2) a controlled, in-lab study of Livefont legibility with both low-vision and sighted participants, presenting a more efficient transcription-based methodology for evaluating legibility without training, and 3) the first evaluation of the learnability of an animated script, with both low-vision and sighted participants. Livefonts support the thesis statement by demonstrating that legibility barriers to accessing textual information can be further addressed by using computer graphics to animate letterforms, through a design process guided by participatory design and optimization over crowdsourced data. Livefonts were initially published in [23].

Chapter 5 Our guiding research questions are: *Can animating an ASL character system help remove barriers to adoption and use? How can we design these animations in a structured way with input from ASL users?* We address these questions through Animated Si5s, the first ASL character system prototype *with animated letterforms*, introducing the potential to increase text resemblance to live signing and consequently improve learnability. Our key contributions are: 1) design and implementation of the first animated ASL character system prototype, 2) an opportunity evaluation study suggesting a need for an animated ASL character system, and 3) the identification of design dimensions for representing sign movements in a character system and guidelines for appropriate designs based on a participatory design workshop with ASL users. In support of the thesis statement, our work on Animated Si5s shows that using computer graphics to design an *animated* ASL character system according to data collected from a participatory design process can help remove barriers to accessing textual information in ASL. The animated ASL character system work will appear in [24].

Chapter 6 Our guiding research question is: *How can we build an ASL-to-English dictionary that is robust to query variability?* In answer to this question, we present ASL-Search, a feature-based crowd-powered ASL-to-English dictionary, and ASL-Flash, a crowdsourcing platform that gathers sign feature evaluations for the dictionary while teaching ASL vocabulary. Our key contributions are: 1) the design of a feature-based ASL-to-English dictionary that handles language and query variability by learning from its users' query data (feature evaluations) through topic modeling, 2) the design and implementation of ASL-Flash, a volunteer crowdsourcing platform that

serves as an ASL educational tool while gathering data to power ASL-Search, and 3) a proof-of-concept demonstrating the viability of our design. ASL-Search and ASL-Flash support the thesis statement by demonstrating how access to sign meanings and sign language education can be expanded by redesigning sign language search to be more robust through crowdsourcing and topic modeling. The ASL-Search and ASL-Flash system designs were originally published in [25].

Chapter 7 Our guiding research questions are: *How well does our ASL-to-English dictionary design perform at scale? How can we change our design to better support scalability?* In response, we explore the scalability of ASL-Search and ASL-Flash. Our key contributions are: 1) a scaled ASL-Flash design that addresses challenges of coverage and incentivization, and 2) a preliminary scalability study of our ASL-Search dictionary design based on data collected through a deployment of the scaled ASL-Flash design. Our scalability exploration further supports the thesis statement by demonstrating that crowdsourcing and topic modeling can support improved sign language search at scale as well. The work in this chapter has not previously been published, but the scaled ASL-Flash design is available at `http://aslflash.org/`.

Chapter 2

RELATED WORK

Historically, written language has been defined by stationary (non-moving) line-drawn characters, and sign language search interfaces have been largely text-based. Our work extends typography to include animation as a possible defining feature of letterforms, and introduces a new crowd-powered search technique for sign language not based on text.

As such, related works spans 1) the psychophysics of reading (Section 2.1), which highlights vision-induced reading challenges our Smartfonts and Livefonts aim to improve; 2) English character design (Section 2.2), a conventionally constrained design space that Smartfonts and Livefonts greatly expand; 3) ASL representation systems (Section 2.3) comprising stationary characters (text) and animated videos, to which we add the first animated ASL character set; 4) ASL lookup, recognition, and translation systems (Section 2.4), to which our ASL dictionary contributes a method for handling query variability; 5) crowdsourced perception studies (Section 2.5), which we use to gather sufficient data to inform, power, and evaluate several of our proposed systems; and 6) participatory design (Section 2.6), which we used to closely involve ASL users in the design process of Animated Si5s.

2.1 The Psychophysics of Reading

Since we are developing alternate ways to render text, it is important to understand how people process text.[1] The impact of visual perception on reading is particularly relevant to our work on improving text legibility. During reading, the eye creates a retinal image of the displayed text, which is subsequently processed by the brain. This first visual step impacts reading and causes

[1] We recognize that there are non-visual forms of reading. For simplicity, "reading" refers to visual reading in this document.

bottlenecks for low-vision readers. This section discusses "typical" reading, low-vision reading, and motion perception as it relates to animated character reading. By aligning with people's visual reading abilities, our Smartfonts and Livefonts are designed to improve legibility.

2.1.1 "Typical" Reading

When people read, their eyes dart from one fixed position to the next in jumps called "saccades." The majority of the time, an experienced reader's eyes are stationary. The region from which a person's eyes can gather information during a fixation is the "perceptual span."

There are several competing models of how people convert visual text into meaning. Word identification lies at the heart of many of these models. For example, dual-route (i.e., dual-process) models (e.g., [30]) propose that there are two ways that people recall word meanings: 1) by sounding out the word's phonemes and registering the word by its sound or 2) by converting the word's visual representation directly to vocabulary. In contrast, single-route models propose that lexicographical, phonological, and orthographic units exist in the brain; in between lie hidden computational layers that refine over time with experience.

The importance of letter identification in reading models informs our focus on improving character recognition in designing Smartfonts and Livefonts. In the phoneme-based route of dual-route models, sequential letter identification is thought to play an important role in sounding out words [153]. The number of letters that fit in the perceptual span has also been linked to reading speed [89], providing further evidence of the importance of individual letter identification.

Text's visual appearance impacts reading in many ways. For example, perceived text size [91] and luminance contrast between letters and background [94] impact reading. Vision is clearest in a narrow central field of view processed by the fovea, while peripheral vision typically has lower acuity. As a result, the number of letters identifiable in the periphery is limited. This limits the number of letters recognizable in a single fixation, which is strongly correlated with reading speed [89, 87]. The reading process is still largely not understood, and both high-level cognitive processes and low-level physical mechanisms involved are active research areas [134].

2.1.2 Low-vision Reading

Low vision is typically characterized by low acuity, making small text illegible and reading slow. Central vision field loss, where the central retinal picture is absent, is also common to low vision, caused by common diseases such as macular degeneration. Central field loss contributes to difficulty reading [48, 163, 92] by forcing people to use peripheral vision to read, which has lower acuity (more "blur").

Consequently, low-vision readers often use strong magnification [87]. Many browsers provide built-in magnification, software like ZoomText [136] supports cross-application magnification, and closed-circuit television (CCTV) systems can magnify and project even more diverse contents onto a screen. Magnification makes it particularly difficult to fit enough content on small screens to use personal devices effectively. Even on large screens, magnification impedes reading by limiting the window of legible text and requiring panning, which can be so cumbersome that many low-vision readers abandon magnification altogether [147]. To help address such problems, Smartfonts and Livefonts use visually distinct characters to compress text while maintaining legibility.

Color can also impact legibility for low-vision readers, especially for vision partial to certain wavelengths [93]. For example, colored lenses are known to ease or speed up reading [166]. White text on a black background is commonly preferred [133, 147, 174] by readers with a clouded lens, which scatters light and creates glare. Because a black background reduces light and subsequent glare, it often improves reading. Due to this general preference, we use a black background for several smartfont/livefont designs and experiments.

2.1.3 Animation and Reading

As animation integrates into text through animated emojis, GIFs, and our proposed Livefonts and animated ASL characters, motion perception becomes newly relevant to reading. The psychophysics of motion perception is an open research area, though some similarities between sighted and low-vision motion perception have been established. The human retina typically has a high concentration of cone receptors (which perceive color) in the fovea, which is responsible for cen-

tral vision, while rods (which perceive light) are more pervasive in the rest of the retina, which is responsible for peripheral vision [122]. Consequently, typical vision is relatively sensitive to peripheral movement, especially when involving light. In contrast, motion detection tends to deteriorate for low vision in the periphery [148]. However, low vision is similarly sensitive to typical vision in central motion detection [148], and in peripheral motion detection with sufficient motion speed [86]. Low vision also exhibits larger variance in motion detection than typical vision. Given the similarities between low vision and typical motion perception, we suspect that animating letters, as we do in Livefonts, might benefit both low-vision and sighted readers.

Animation can distract readers, but according to past studies, the effect depends on the design of the animation and the overall display. There is evidence that animations unrelated to text being read can negatively impact a reader's ability to find desired content, though the animation design and difficulty of the search task can mediate this effect [121]. Flashing displays in particular can attract viewers' attention, but the level of distraction is mediated by the extent to which other items on the screen attract the viewer's attention [68]. Other work suggests that appropriately designed peripheral animations can enhance text displays without significantly distracting users [121]. It is possible that peripheral animations in Livefonts and Animated Si5s will distract readers. However, viewing modes that animate only characters being read (e.g., through eye-tracking or hovering over desired content) or fairly uniform patterns of animation throughout longer passages of animated text could eliminate or reduce such distractions.

2.2 English Character Design

Like other traditional alphabets, English characters have evolved to support both visual reading and manual writing. As described in this section, fonts provide some variability in character design, alternate English scripts like Braille set a precedent for completely redesigning English letterforms, and most recently animated emoji have paved the way for mainstream animated text. Unlike traditional alphabets, Smartfonts, Livefonts, and Animated Si5s are designed for display on screens and thus are freed from the constraint that they be easily written by hand. Removing this constraint allows Smartfonts and Livefonts to improve legibility beyond what is possible with traditional

alphabets, and Animated Si5s to introduce iconic motions to ASL text.

2.2.1 English Font Design

Font design impacts the reading experience. Various letter shape properties, including stroke width or boldness (e.g. [12]) and serifs (e.g. [5]), have been shown to impact legibility. Contrast level in both luminance [95] and color [90] can impact both readability and aesthetic appeal. Certain text/background color combinations are known to be more readable and pleasing than others [61, 53, 130]. Prior studies on color-grapheme synethesia, where people have strong associations between letters and colors (see, e.g., [29]), have shown that reading books with colored letters suffices to learn and create strong perceptual associations between letters and colors, without active attention to text color. There is also evidence that named colors are more easily recognized [167], so our Smartfonts and Livefonts use colors with distinct English names. Font design can address various challenges including dyslexia [124], aviation [157], and low-vision reading, described next.

Several fonts have been designed for low vision (e.g., Tiresias [57] and APHont [50]). To explore whether novel scripts can improve over the "best" traditional letterforms, we compared Smartfonts and Livefonts to a state-of-the-art low-vision typeface, Matilda [16, 15]. The typeface was designed by Bessemans for low-vision children, who are at a disadvantage compared to their sighted peers. It is characterized by "wide, open, and round letters" with a "friendly feeling" [15]. Based on both structured experimentation and design experience, its design is both scientifically rigorous and aesthetically pleasing. Due to the wide diversity of low-vision conditions, font personalization is also particularly effective for low vision, as evinced by the wide array of fonts low-vision readers created for themselves when given the chance [4]. Smartfonts and Livefonts introduce a larger, more flexible design space for personalized and general low-vision scripts.

2.2.2 Alternate English Scripts

Smartfonts and Livefonts are conceptually similar to Braille, a tactile writing system for people who are blind or have very low vision. In Grade 1 Braille, each English letter is represented by a

collection of raised dots in a 2x3 grid, replacing traditional embossed letterforms. Because Braille completely redesigns character shapes, it faced opponents who thought it too radical, unhelpful or detrimental to the blind community, and even attempted to ban it [52]. Despite initial resistance, Braille was eventually accepted, and has greatly benefited many people, with usage linked to higher rates of employment, education, and financial stability [127]. Like Braille, Smartfonts redesign the written form of each English letter to improve legibility. Some of our designs even adopt Braille's 2x3 structure. Tactile sensing of letters is similar to visual sensing of letters subject to a low-pass spatial filter (i.e. blur) [101]. This suggests that, just as Braille's 2x3 structure is more legible than traditional letters to the touch [100], a 2x3 structure might improve legibility for readers with blurry vision.

There is also a tradition of unconventional visual alphabet design predating computer screens. Of particular interest is Green-Armytage's response [58] to Rudolf Anheim, a prominent perceptual psychologist who asserted that an alphabet differentiating letters through color would be unusable [6]. Green-Armytage compared alphabets comprised solely of different colors, shapes, and faces, and found the color alphabet to be identified most quickly. The idea that constraints on effective alphabet design are looser than we intuitively think is supported by psychophysical models of visual letter recognition being accomplished through recognition of simple features [118, 54]. Consequently, "any set of characters should do, as long as they contain a sufficient number of simple, detectable features" [87].

2.2.3 Animation and Text

Animation, which we propose adding to English and ASL scripts, has already become part of reading on digital devices, as animated emoji and GIFs have integrated into text. Emoji [111] are pictures or animations rendered by text applications. Emoji are internationally popular and used for a variety of purposes [76]. They offer richer displays than their predecessor, emoticons, low-tech pictures made of keyboard characters (e.g., :-D). Recently, providers started animating emoji to create even more engaging text displays. Our animated scripts further this trend of enhancing text through animation.

The successful integration of animated emoji in text demonstrates the technical feasibility of Livefonts and animated ASL characters. A growing Unicode block is reserved for emoji [38], supporting smoother cross-platform rendering, and underscoring their prevalent use. Applications support different renderings of these Unicode characters, and some expand upon this standardized set. For example, Skype provides animated emoticons like a hugging bear, and story-like GIFs they call "Mojis"; and Facebook inserts animated stickers into chat conversations. SMS applications are starting to offer similar animated options. Animated GIFs are also integrated into text-based social media like Twitter and Facebook and used by electronic newspapers[2] to introduce or enhance text-based articles. With increased support for animated emoji, Livefonts and animated ASL characters will become similarly technically feasible, and extend this trend towards rich, animated text interfaces.

2.3 ASL Displays

Displaying content in ASL is important for preserving Deaf culture, making content accessible to ASL users who might not know English, and supporting expression in ASL. ASL has been documented and displayed in two ways: 1) stationary ASL character systems, and 2) dynamic ASL videos. This section outlines both approaches, which inform our proposed animated ASL characters, a hybrid providing the functionality of text while portraying sign movements directly.

2.3.1 ASL Writing Systems

There are two main types of ASL writing systems: transliterations, which use another language's words but preserve ASL grammar, and character systems, which present a set of symbols used according to a set of rules to represent the language. ASL character systems can be further organized into three main groups: those for 1) linguistic analysis, 2) everyday use, and 3) computer modeling of signs. Examples of each are provided in Appendix Table A.1. All past writing systems are designed for physical writing, and are consequently static, making it difficult to represent movement

[2] e.g., `http://www.nytimes.com/2017/03/10/opinion/sunday/can-sleep-deprivation-cure-depression.html`, accessed 2017-03-13.

iconically. None have secured mass adoption, facing the barrier of time and effort required to learn a new system's symbols. Animating ASL characters, as we propose for the first time in Animated Si5s, increases notational resemblance to live signs, potentially removing this learning barrier.

Gloss refers to transliterating one language with another. ASL transliterations are commonly provided through English gloss, which represents each sign with an English word in all-caps (e.g., BOOK). Fingerspelling is written with dashes (e.g., R-I-C-H-A-R-D), and additional symbols or descriptions provide details according to various glossing conventions (e.g., [145]). Gloss is popular for teaching ASL to hearing students who already know English, because it uses familiar English words yet preserves ASL grammar. It has also been argued that gloss can help Deaf children learn English [144]. However, gloss leaves out many sign details, provides only limited support for representing parallelism, and its reliance on English undermines ASL's status as an independent language. It also does not represent sign movements iconically, as our animated ASL character system can.

ASL character systems for linguistics are typically extremely detailed, which can make them difficult to learn and use. They are typically both developed and used by linguists, and document details of the language important for linguistic analysis not necessarily useful for everyday conversation. Stokoe performed the initial linguistic analysis of ASL that helped establish ASL as a full natural language, decomposing it into five types of features – handshape, location, orientation, movement, and relative position [138]. Stokoe notation, which is based on this featural decomposition, is ASL's seminal linguistic notation system [139]. Subsequently proposed linguistic notation systems are largely based on Stokoe notation (e.g., [27, 77]).

Like linguistic notation systems, systems for modeling signs on computers provide details about body movements not necessary for everyday communication. Many are based on linguistic notation systems, including HamNoSys [63], one of the most popular notation systems used in research [107]. One use is digital transcription of linguistic data to create sign language corpora that can be searched and analyzed (e.g., iLex [62]). Another common use is the internal representation of signs for translation systems, which often output the translation through a signing avatar. Markup languages (e.g., SigML [46]) store the notation in a way that is accessible to the anima-

tion software. Beyond sign language, HamnNoSys and several variants are popularly used among motion researchers to transcribe motions (e.g., [83, 103, 56, 66]).

Character systems for everyday use aim to support reading and writing. Many of these systems were proposed by individuals to fulfill a need to write in ASL (e.g., SignWriting [146], si5s [10], SignFont [112], ASL-phabet [143], ASL Orthography, SLIPA [119], and ASLSJ [141]). Unlike linguistic or computer notation systems, they capture high-level information to convey meaning without cumbersome linguistic details. Like all writing systems, ASL writing systems abstract away some of the live language, and vary in captured aspects and visual presentation. Many are featural, meaning that sign features are represented by separate symbols, which in combination represent signs. Many are iconic, meaning that the symbols visually resemble signs. Some of these notation systems are not sequential, instead allowing characters to be positioned in two dimensions, to account for structural differences in ASL compared to sequential languages like English [165]. Because these systems were designed for traditional paper rendering, they are limited in capturing motion iconically. By incorporating movement in characters, we introduce the possibility of representing sign movements more iconically in text. Our animated ASL character system prototype builds off of si5s, a fairly iconic ASL character systems.

Technical support for ASL character systems is developing. Keyboards (e.g., the SignWriting Keyboard [3]) have been proposed to support writing in ASL on the computer. These keyboards are often inconvenient to use because character systems can have more than one degree of freedom in placing symbols on the page, requiring dragging or positioning with arrow keys rather than single-strike typing. Because ASL character systems can have many more than 26 characters, mapping these symbols to the QWERTY keyboard is also a challenge. SignWriting has received official recognition in Brazil for Deaf education [132] and a set of Unicode characters have been reserved for SignWriting [31] to provide cross-platform support, signifying the growing demand and use of ASL writing. However, traditional Unicode scripts support sequential languages like English, not nonsequential scripts like SignWriting. Solutions such as SWML, an XML-based markup system,

[3] `https://swkb-35431.firebaseapp.com/`

have been proposed to address this problem [34]. SMS support has also been explored [1]. No fonts are available for installation, though si5s characters are publicly available for download[4]. Using animation to make ASL characters more immediately accessible on screens furthers this trend towards computer usability.

2.3.2 ASL Video

ASL video systems use recordings of people or animated signing avatars to display content in ASL. Because ASL videos capture movement, they can resemble the movements of in-person signing more closely than stationary ASL characters. While ASL video is a natural, powerful medium for ASL [72], it is not always an appropriate substitution for text, as it does not provide the same functionality. Unlike text, video presents content through a specific body rather than abstractly, does not easily integrate into text-based platforms, and does not readily meet other text-met needs, e.g., discrete note-taking and iterative, collaborative content generation. By animating a character set, we provide both the dynamic expressiveness of video and the functionality of text. We do not seek to replace ASL video, rather to create a more iconic form of ASL text.

ASL video recording is used for both real-time communication and fixed content display. For example, video relay services (VRS) allow deaf and hard-of-hearing people to converse with hearing people over the phone, by going through an interpreter via video. Video calls also supports direct communication in ASL, through platforms such as Skype and Google hangouts. However, live video-based communication requires participants to sign ,which can draw unwanted attention, and a video camera, which can be expensive or unavailable. Some interfaces use ASL recordings to present fixed content in ASL, in particular educational tools. Examples include the ASL-STEM Forum [17], a platform for creating new STEM signs; MTN [44], a program to teach sign language; ASL-phabet [143], a signing dictionary for children that not only demonstrates signs in ASL, but also uses sign videos in the interface itself; AILB, an e-learning platform [142]; and systems to teach English literacy in parallel with ASL [64]. These platforms have been shown to increase

[4]Source: `http://aslized.org/innovations/si5s/font-download/`

content accessibility and user satisfaction.

ASL avatars enact signs through the body of a cartoon character on a screen. To manipulate the avatar, a low-level description of sign movements is created, which is then processed by the graphics component responsible for animation. The low-level representations are stored in detailed notation systems such as HamNoSys [63] (discussed above) and its variants (e.g., [78]). Accurate models of human movement are required to generate life-like movements from the notation (e.g., [71, 169]). Content for avatars to sign is typically generated either by translating English text into ASL, or by recognizing signs executed by a live signer. Researchers have attempted to translate English into signed ASL (e.g., [173, 70, 109]), though none of these systems work completely. Researchers have also attempted sign language recognition, but their systems typically fail to generalize or scale (as described earlier in the related work). Consequently, a human-in-the-loop approach is currently needed to animate ASL avatars reliably.

Replacing text with sign videos to improve accessibility can introduce new interaction problems. Unlike replacing one written language with another, replacing text with video changes the layout. For example, replacing individual words with videos can expand the interface in inappropriate places. Similarly, replacing longer text with video can collapse the interface in undesirable ways. Providing an interface that supports both text and video to accommodate all users is also challenging. To solve these problems, researchers have proposed a variety of strategies, including strategically layering sign language videos on top of existing web designs [39]; and creating customized tooltips with sign videos or pictures [120]. However, these solutions do not provide a truly equal and equivalent experience for both ASL and non-ASL users.

2.4 ASL Lookup, Recognition, and Translation

Sign language lookup, recognition, and translation systems accept two types of input: 1) a physical demonstration of the sign, or 2) a set of descriptive features. Query by demonstration requires expensive hardware, and relies on gesture recognition algorithms that cannot yet translate signs accurately. Feature input simplifies the translation problem and gives users flexibility, but existing feature-based systems do a poor job of matching input features to signs. Our ASL-to-English

dictionary accepts feature descriptions of signs from users and learns from these descriptions, allowing for diverse search queries without additional hardware.

2.4.1 Query by Demonstration

Systems that query by demonstration must capture the physical sign demonstration. Three common input mechanisms are video, 3D cameras, and motion capture gloves, each described below.

Recognizing signs from video is difficult [32, 73, 114, 137, 160, 170]. To recognize nuanced gestures, video-based systems often track pixel color (e.g. [73]) using Hidden Markov Models (e.g. [137]) or Time Delay Neural Networks (e.g. [170]). Because these methods typically extract low-level features, while humans use high-level visual features [126], computational methods have not yet caught up to human sign recognition. Some sign identification systems achieve high accuracy when trained and tested on the same users, but make no performance guarantees when trained and tested on different users (e.g., [160, 32]). Systems that can handle diverse users are typically only tested on small corpora and make no scalability guarantees (e.g., [114]), though recent statistical methods show promise for scalability (e.g., [82]).

More recently, 3D cameras such as the Kinect[5] and Leap Motion[6] have been explored for sign language recognition [45, 171]. These cameras support skeletal tracking of joints in the hands and body in 3 dimensions. As in 2D video recognition, HMMs are commonly used to model these skeletal gestures over time (e.g. [171]), and researchers have attempted to overcome similar difficulties in generalizing to diverse users (e.g., [45]). While companies do not publish their algorithms, commercial products using 3D cameras for sign language recognition (e.g., MotionSavvy[7] and SignAll[8]) seem to have comparable limitations. Even with 3D tracking data, accurate sign language translation for diverse users with a complete sign vocabulary is an open problem, complicated by dependence on the quality of the user's sign demonstration and computationally

[5] http://www.xbox.com/en-US/xbox-one/accessories/kinect

[6] `http://www.leapmotion.com/`

[7] `http://www.motionsavvy.com/`

[8] `http://www.signall.us/`

expensive data processing.

Gloves and other arm and hand sensors have also been used for sign detection [80, 162, 98, 81]. While many of these systems were not designed for sign language specifically, they support detection of nuanced hand movements and positions necessary for sign recognition, for example providing depth detection [80] or individual finger movement tracking [162]. However, these systems require specialized hardware that can be expensive and uncomfortable.

2.4.2 Query by Feature Selection

Feature-based ASL-to-English dictionaries, like ours, avoid the computational problems of parsing sign demonstrations by allowing users to input features describing a sign. This subsection reviews printed and electronic feature-based lookup resources, outlines their challenges, and describes Latent Semantic Analysis (LSA), an indexing and lookup method adapted by our dictionary to improve search results over existing resources.

Printed ASL-to-English dictionaries organize signs by linguistic features, taken from linguistic notation systems like Stokoe notation [139]. These dictionaries sort signs first by a single feature such as location or handshape (e.g., [140, 150]). If the user does not know this first feature, the dictionary cannot help them find the sign. Even when the feature can be identified, users still need to sift through many possibilities. Our dictionary uses features from Stokoe notation, but handles mistaken, forgotten, and subjective feature inputs.

Electronic feature-based ASL-to-English dictionaries allow the user to select features to describe a sign (e.g. Handspeak[9], Jinkle[10], SLinto[11], and The Ultimate ASL to English Dictionary (TUAED) [154]). Once the user selects a set of features using the search interface, the system returns possible matching signs if any are found. Existing feature-based ASL-to-English dictionaries have several limitations:

---

[9] http://www.handspeak.com

[10] http://asl.jinkle.com/lsearch.php

[11] http://slinto.com/us/

1. **Poor matching of features to signs.** - These dictionaries do not seem to be fully functional, and do not publish their lookup algorithms. Based on experience using these systems, it is likely that they find matching signs by executing an exact database lookup.

2. **Requirements on features that the user must select.** - Handspeak requires hand shape, movement, and location for one unspecified hand; Jinkle requires the starting hand shape, orientation, location, and movement for the dominant hand; and SLinto requires hand shape and location for both hands.

3. **Lack of support for feature omissions.** - Only SLinto allows for missing features.

4. **Cumbersome search interfaces.** - Many interfaces are English- not ASL-oriented, requiring feature selection through drop-down English word lists (e.g., Handspeak, Jinkle), or returning results sorted alphabetically by English meaning (e.g., TUAED).

Our dictionary design addresses each of these challenges with a front end that gives the user freedom in feature selection (limitation 4), and a backend that learns from patterns and variations in user queries (limitations 1-3).

2.5 Crowdsourced Perception Studies

Crowdsourcing provides a powerful tool for gathering perceptual data, which we use to generate data for several of our proposed systems. Crowdsourcing has been shown to yield reliable results for perception studies, and has been used by many researchers. For instance, Demiralp et al. explored the use of crowdsourcing to evaluate the perceptual similarity of different shapes and colors, and developed perceptual kernels to quantify crowd-learned similarity [41]. They found crowdsourcing to be an inexpensive, rapid, and efficient means to gather data on human perception. Heer and Bostock demonstrated the viability of Mechanical Turk, a popular crowdsourcing platform, for evaluating visualization graphics by replicating previous results and running new studies to produce new insights [65]. In each of our three proposed projects, we use crowdsourced

perceptual data to design solutions, using feature evaluations to build our ASL dictionary, mistaken character identification to inform smartfont and livefont designs, and tuned animation parameters to inform ASL character animations.

Crowdsourcing has been shown to produce comparable results to in-lab studies [69, 116, 55], which we take advantage of to evaluate our scripts with larger populations. In-lab evaluations of text readability and legibility typically require participants to read text and then complete a derivative task, such as a basic comprehension test (e.g. [61]). Reading time and comprehension level serve as metrics for readability. An alternate setup presents a paragraph of text with individual word substitutions (e.g. [14, 36, 13]). The number of word substitutions detected measures readability or legibility. Other tests involve showing a single word or pseudoword, and asking the person whether the word they saw was a real word (e.g. [42]). Accuracy in distinguishing words from non-words in relation to word display time determines legibility. Pelli et al. [118] compare "efficiency" of letter identification across traditional and made-up alphabets, measured by how well individual letters were identified in the presence of random noise. They also found that a few thousand training examples sufficed to teach someone to identify unfamiliar letters fluently. We bring similar evaluations of smartfont legibility and learnability to the crowd to bring structured evaluation to a larger pool of potential users and gather more data.

Crowdsourcing is particularly scalable when tasks align with the crowd's incentives. Examples of non-monetary compensation for crowd work include education (e.g., Duolingo [158]), scientific contribution [131, 20], personalized recommendations [117], self-discovery (e.g., LabintheWild [123]), fun (e.g., Foldit [33] and the ESP game [159]), and community (e.g., Wikipedia[12] and Stack Overflow[13]). By providing tasks with inherent value, these platforms often create long-lasting benefits and worker involvement. A self-motivated crowd also does not require monetary compensation, which makes these systems sustainable. By serving as an educational resource, ASL-Flash, our proposed platform for gathering ASL feature data, is similarly scalable.

---

[12] https://www.wikipedia.org/

[13] https://stackoverflow.com/

2.6 Participatory Design

Participatory design is a methodology for involving all stakeholders, meaning everybody affected by a technology (or other product), in the design process. Participatory design (originally co-operative design) empowers users by giving them an equal voice in the design process [49, 19]. Methods include interviews and observations, low- and high- fidelity prototyping, design sessions, structured brainstorming, and workshops (as described in [129]), and more recently distributed methods to involve remote users online (e.g., [60, 35]).

Participatory design has been used in a range of fields beyond technology, and has a rich literature spanning politics, participation, and methodology (as described in [79]). In particular, people with disabilities have been successfully involved in designing accessible solutions through participatory design (e.g., people with dementia [99], children with special needs [51], and Deaf students [172]). We used participatory design principles to involve Deaf and hard-of-hearing ASL users in the design process for our animated ASL character system, Animated Si5s. Involving members of the Deaf community in the design was particularly important for this project, as ASL is the primary language of the Deaf community.

Chapter 3

SMARTFONTS

Can we improve text legibility by leveraging computer graphics to redesign our letterforms? How can we design these alternate scripts in a principled way?

Text-based resources are pervasive, yet present information access problems due to poor legibility. The Latin letterforms we use today (the familiar A-Z) are constrained to symbols that are easily hand-written. However, modern graphics support richer letterform designs (e.g., involving color and shapes that are not line drawings) that might help optimize letterform legibility. In light of this opportunity, we propose redesigning our letterforms from scratch, leveraging the display capabilities of modern screens to increase legibility. These novel scripts, called *Smartfonts*, aim to improve the reading experience by replacing traditional letter shapes with more easily distinguished characters. We focus on English, but similar ideas can be applied to other languages.

Smartfonts, demonstrated in Figure 3.1, comprise distinct renderings of the twenty-six Latin letters used in English so users can read text, letter for letter, without changes in orthography. They do *not* involve spelling changes or shortenings such as reading without vowels, though these

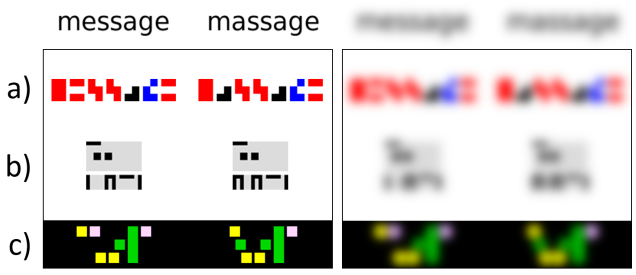


Figure 3.1: The words *message* and *massage* clear (left) and blurred (right) in our Smartfonts: (a) Tricolor (b) Logobet (c) Polkabet (on black). Words are sized to have equal areas.

Figure 3.2: A smartwatch displaying an SMS. The sender is oblivious to the fact that the SMS is read in a Smartfont.

tactics could be used in combination with Smartfonts. Software can render text in Smartfonts as easily as existing fonts. For instance, we have modified the firmware of a smartwatch to display everything in Smartfonts (see Figure 3.2). Importantly, Smartfonts could be used by any individual to display any digital text *without large-scale adoption*. Secondary potential benefits include increased privacy (e.g., for personal messages that may pop up), improved reading speed, aesthetics, personalization, and comfort (e.g., reduced fatigue).

We designed five Smartfonts to improve legibility of small, blurry text commonly viewed on small screens, and optimized our designs by iteratively evaluating them with crowd workers. Our Smartfonts employ blocks (not merely strokes) of color to preserve character distinguishability under blurry reading conditions. Blurring replaces each pixel with a weighted mixture of surrounding pixels. Blocks of color survive largely unchanged because nearby pixels already share the same color. A diverse color palette also helps differentiate between confusable characters. These techniques produced Smartfonts rendered with high fidelity at very small sizes, some perfectly renderable at only six pixels per letter.

We evaluated Smartfont legibility and learnability with crowd workers. Because the general

public does not know how to read Smartfonts, evaluating their legibility is difficult. Our methodology does not require training, instead asking crowd workers to identify (match) random strings in our Smartfonts and in Latin characters. Blurry vision was simulated by applying a Gaussian blur to the text, and size was varied. We also evaluated the difficulty of learning to read Smartfonts and observed a learnability/legibility trade-off. We present evidence that our most learnable Smartfont (in Figure 3.1a) can be read at roughly half the speed of Latin after two thousand practice sentences. It is also legible at smaller than half the size of traditional Latin when blurry.

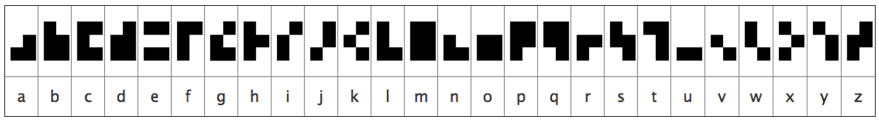
Our key contributions are: 1) introducing the concept of Smartfonts that radically redesign characters to improve certain aspects of the reading experience, 2) demonstrating how one can design and optimize (based on data) Smartfonts for learnability and legibility under specific reading conditions, in our case, small blurry text, and 3) providing a methodology to evaluate legibility without teaching people to read fluently.

3.1 Our Smartfonts

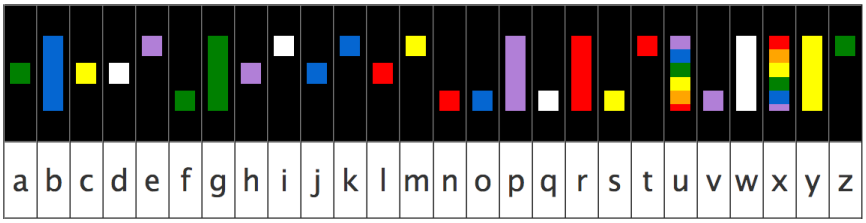
We designed three initial Smartfonts to be easily legible at small sizes and out of focus. We leveraged three main techniques: 1) using blocky shapes known to be resilient to blur, 2) using color to distinguish between characters, and 3) radically reducing the space between adjacent characters. Visibraille and Polkabet are designed to support blurry character distinguishability, while Logobet is designed to minimize text area.

3.1.1 Visibraille

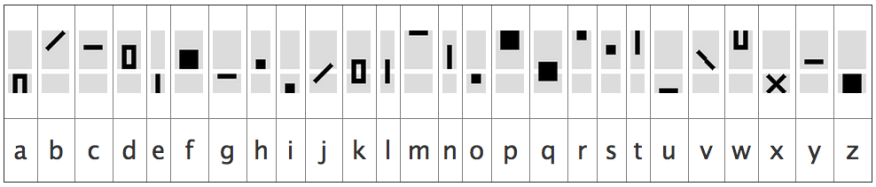
Our Smartfont Visibraille is modeled after Braille. Braille's 2x3 structure of dots has proven to be more tangible than embossed traditional letter forms [100]. Because fingers have low spatial resolution, what is "seen" with the fingers resembles what is seen in a blurry image [101]. Consequently, we expect Braille's 2x3 structure to be more easily discernible than traditional characters for blurry vision. This expectation is supported by empirical evidence showing that 2×3 characters are more visually distinguishable than traditional characters; out of a range of established and



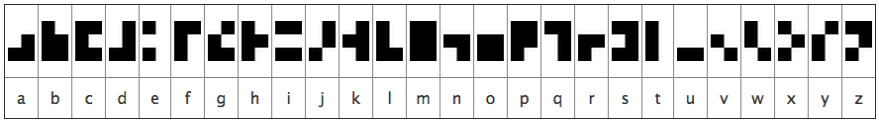
(a) Visibraille alphabet



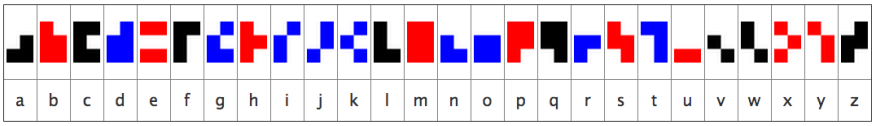
(b) Polkabet alphabet



(c) Logobet alphabet



(d) Visibraille 2 alphabet



(e) Tricolor alphabet

Figure 3.3: Our Smartfonts designed to improve legibility of small, blurry text.

made-up alphabets, 2x3 characters have been shown to be most visually recognizable [118].

Visibraille, shown in Figure 3.3a, maps 26 2x3 blocks onto the 26 letters of the English alphabet. We selected the 2×3 blocks to be similar in shape to the English characters they represent. Because of its simple design and similarity to Latin characters, we expect this Smartfont to be both legible and learnable.

3.1.2 Polkabet

Polkabet adds color to the design space to increase distinguishability. It is designed for a black background, which is common to personal devices like the smartwatch. Contrast in color and luminance help the eyes distinguish elements visually. They impact text readability (e.g. [95, 90]), and help data visualizations distinguish between data (e.g. [161]). To support distinguishability, we chose five colors, shown in Figure 3.3b, spaced out in hue and to have a strong luminance contrast with the black background. We then adjusted them based on participant feedback, since colors vary greatly across displays. Similarly, colors could be personalized and tailored for various types of color blindness. Rather than adding an additional color, we included a rainbow of the other hues as a possible character coloring, to minimize the number and confusabililty of colors.

Polkabet's characters are solid squares and rectangles. Perimetric complexity (the ratio of character perimeter to "ink" area) has been found to correlate strongly with people's efficiency at identifying characters, with less complex characters identified more efficiently [118]. Because solid squares and rectangles have low perimetric complexity, we expect these shapes to be highly discernible. Characters with low perimetric complexity are also resilient to blur. When an image is blurry or out of focus, each pixel appears to be a mixture of nearby pixels. Solid blocks are highly robust to this type of blurring because many nearby pixels share the same color.

Like Braille, Polkabet uses dot position to distinguish certain letters. While many letters are similar and hence possibly confusable, the similarities are vertical, motivated by the preference for vertically reflected letter pairs such as p-b over horizontally reflected letter pairs such as b-d across natural language [164].

To facilitate learnability, we used mnemonics to match letters to colors, shown in Figure 3.4. If

Figure 3.4: Mnemonics for Polkabet's small square characters.

a character uses a small square of color, the reader can think of the associated item from Figure 3.4. The first letter of that item is the letter that the character represents. Squares of color at the top are associated with foods, middle squares with animals, and bottom squares with miscellaneous items. For example, suppose a reader forgets what a red square at the top of the line means. He/she would think, "This character uses a red square at the top, so think of a red food... Tomato! 'Tomato' starts with *t*, so that's a *t*!" Tall blocks of color represent the first letter of that color.[1]

3.1.3 Logobet

Logobet aims to minimize text area by reducing the spacing between letters, a typographic process called "kerning." In a proportional font, the space allotted to characters depends on their size. For example, an *m* is allotted more horizontal space than an *l*. However, reducing the spacing between pairs of characters can be desirable. For example, a capital *T* allows a short subsequent letter, such as an *o*, to shift left under the *T*'s umbrella, as in the word *Tomato*. Because Logobet employs extreme kerning to condense strings, it visually resembles a character-based logography, such as Chinese.

Logobet's characters were chosen with vertical positioning, like Polkabet, to avoid left-right mirror pairs. Horizontal, vertical, and diagonal lines and circles and semi-circles are known to be

[1] X, which stands for "Rainbow", and U, which stands for "Upside-down Rainbow", are exceptions.

(a) The alphabet in Logobet characters without kerning

(b) The alphabet in Logobet with kerning

Figure 3.5: Example of aggressive kerning with the Logobet alphabet.

easily distinguished [155]. Logobet's characters, illustrated in Figure 3.3c, consist of these shapes except that boxes were used in place of circles to avoid pixelation effects. Logobet maximizes kerning by allowing characters to shift entirely underneath preceding characters. Each character occupies a fraction of the row's height, and subsequent letters that are strictly lower slide underneath. Letters are ordered first top-to-bottom, then left-to-right, to ensure that every printed Logobet string corresponds to a unique letter sequence. For example, Figure 3.5 shows Logobet's kerning on the alphabet.

We hypothesized that Logobet's text compression could improve legibility for low-vision readers by increasing the amount of text that fits legibly on a screen. Kerning reduces the space that each word occupies without reducing the size of individual letterforms, and letterform size is important for identification. With extreme kerning, words also begin to resemble single units or logograms, whose overall shape can be used for identification. However, there is a trade-off between word compression that can thus boost legibility, and decreased space between individual letterforms in a word which might decrease legibility. Our design allows for us to explore that trade-off.

3.2 Optimizations

We explored optimizing color and shape over 2x3 characters. We present two optimized fonts: Visibraille 2, whose 2x3 blocks are chosen to be minimally confusable; and Tricolor, which adds color to Visibraille's Latin-esque 2x3 characters. We generated a crowdsourced confusion matrix

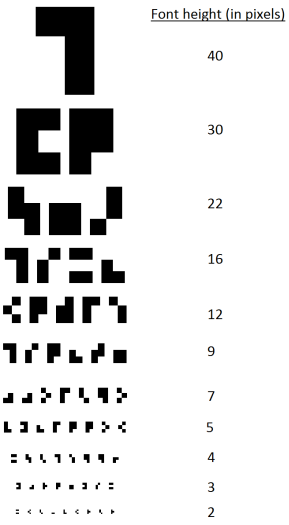


Figure 3.6: Our confusion matrix was generated by testing a series of rows of random characters, like a Snellen chart.

of 2x3 shapes which determined Visibraille 2's shapes and Tricolor's colors.

3.2.1 2x3 Character Confusability Study

We determined the confusability of 2x3 characters[2] using paid crowdsourcing on Amazon Mechanical Turk.[3] Our study is modeled after a rich history of studies on character recognition and legibility, where a predominant technique is to collect confusion information on characters presented in conditions that obscure distinguishability (e.g., [3, 7, 18, 152, 84, 101]), and brings this tradition to the crowd. Our experimental setup mimicked a Snellen eye chart test, a standard eye exam test.

Workers were shown rows of characters at decreasing sizes, specified in Figure 3.6. Each row

[2]Of the $2^6 = 64$ possible 2x3 characters, we considered a subset of 42 characters to reduce labor. In particular, two configurations were considered to have the same shape and likely to be confused if the block patterns were translations of one another.

[3]`http://www.mturk.com`

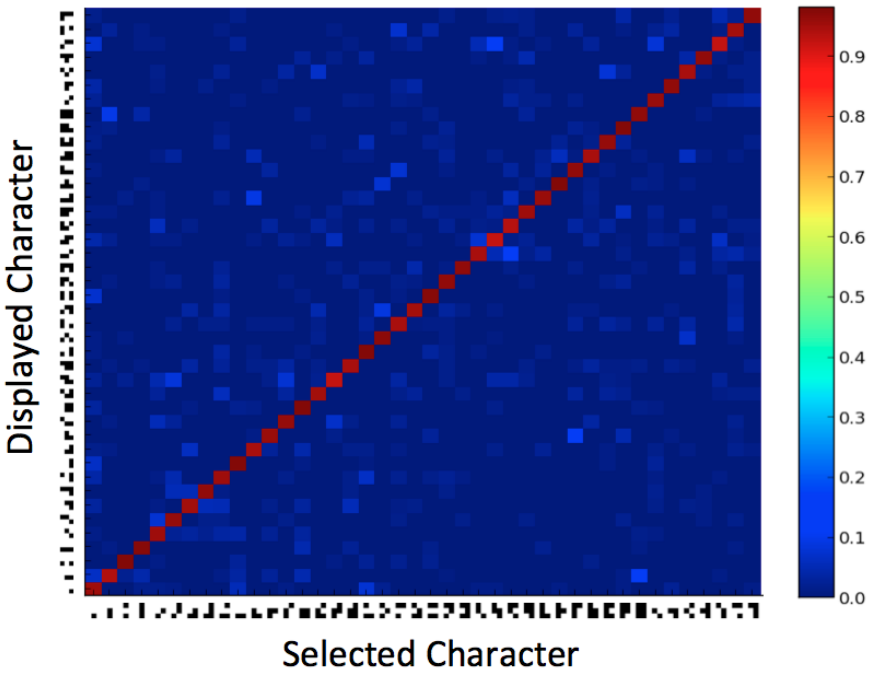


Figure 3.7: Our confusion matrix of 2x3 characters - a matrix showing the frequency with which each displayed character was identified as every other character.

was shown separately, and contained 1-9 characters. Participants transcribed the character(s) using a virtual keyboard of 7 characters. The target characters were chosen randomly with replacement from the keyboard's characters, which were also chosen randomly.

In total, we collected 4022 evaluations from 548 people. Each character was shown 379-500 times (mean 442.1). Each pair of characters appeared together on the virtual keyboard at least 33 times (mean 64.7). Since experiments were conducted remotely through web browsers, we did not control for display conditions or viewing factors such as retinal angle. However, this enables us to assess the confusability of our shapes "in the wild," across a wide variety of display types and people. To minimize pixelation artifacts, participants were instructed to keep their web browsers at the default 100% zoom.

Our confusion matrix C, shown in Figure 3.7, consists of the confusability score c_{ij} between each pair of shapes i and j. Here, c_{ij} denotes the fraction of times shape j was transcribed when i

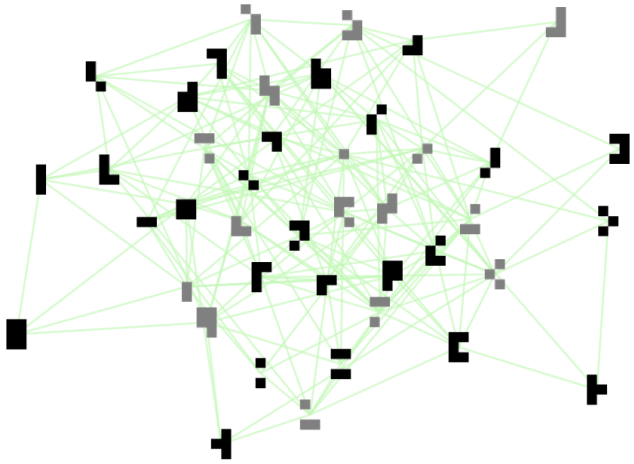


Figure 3.8: The 26 selected shapes (black) of the 42 considered. Although the confusion matrix is high-dimensional (as measured by eigenvalues), D3's force-directed graph layout [21] displays many confusable pairs near one another. Edges denote confusable pairs, which repel each other in the physic simulator creating the force-directed layout.

was the target, out of the total times j was available as a transcription choice for i. The confusability of a shape with itself c_{ii} is similarly defined to be the fraction of times that i was transcribed when i was shown.

3.2.2 Visibraille 2

Finding the 26 most empirically distinct (least "confusable") characters for an optimized Smartfont was modeled as choosing the set S of size 26 so as to minimize $\sum_{i,j \in S, i \neq j} c_{ij}$. This problem is NP-hard (see Appendix A.1 for a proof), but we used a branch-and-bound search to quickly find the exact optimum among the $\binom{42}{26} \approx 10^{11}$ possible solutions. The set found by our branch-and-bound algorithm is visualized in Figure 3.8. The 26 selected letters, shown in black, minimize the sum of edge weights between nodes in the selected letters. We mapped these 26 shapes to the Latin A-Z to create Visibraille 2, as illustrated in Figure 3.3d. The mapping was chosen to ease learning by

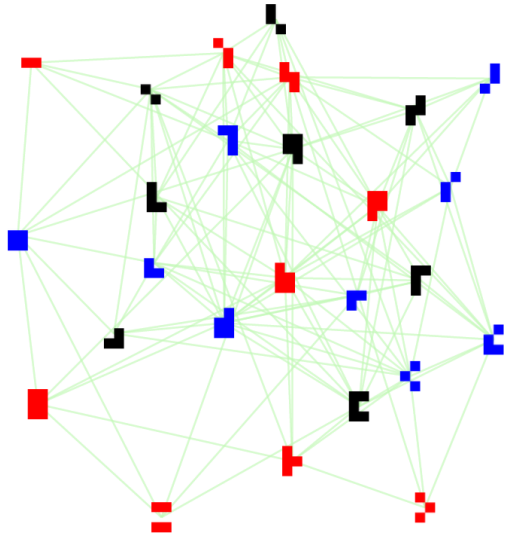


Figure 3.9: A D3 [21] force-directed layout of Tricolor attempts to place similar pairs of letters near one another, and also illustrates the color assignments from our optimization algorithm. Edges denote commonly confused pairs, which repel each other in the force-directed layout.

assigning shapes to Latin characters they resembled (upper- or lower-case[4]).

3.2.3 Tricolor

We also used the confusion matrix to identify commonly confused characters from our initial Smartfont Visibraille. We color each pair of confused characters differently to introduce a new Smartfont Tricolor. Because its characters closely resemble the Latin A-Z, we hypothesized that it would be easy to learn and remember. Because it uses both shape and color to distinguish between characters, we hypothesized that it would also be highly legible at small sizes and blurry.

To assign characters optimally to three distinct colors, we solve the following problem: partition the set of letters into three disjoint sets $S = S_1 \cup S_2 \cup S_3$ so as to minimize $\sum_{k=1}^{3} \sum_{i,j \in S_k, i \neq j} c_{ij}$. This NP-hard search (see Appendix A.2 for a proof of NP-hardness) over $\approx 4 \times 10^{11}$ partitions

[4]For simplicity, we focus on only twenty-six letters, while the Latin font has 52 letters if one includes both cases. An additional twenty-six letters could be added similarly, or like Braille, a single symbol could be incorporated to indicate case.

succumbed easily to exact optimization again using branch-and-bound search. The selection, visualized in Figure 3.9, minimizes intra-color edge weights (between characters of the same color), or equivalently maximizes inter-color edge weights.

The Tricolor alphabet is shown in Figure 3.3e. It adopts the easily-learned Latin-esque shapes of Visibraille, and assigns 3 colors to help distinguish easily confused letters. Because pairs of "mirror images" were commonly confused in our study (in accordance with prior work [164]), our optimization assigned different colors to each such pair. For example, "a" and "n" are mirror images of one another and are colored black and blue respectively. Other highly confusable pairs, such as "l" and "n," are also colored differently.

3.3 Legibility Evaluation

Evaluating the legibility of new Smartfonts is difficult when nobody knows how to read them. Evaluating legibility compared to Latin is further complicated by the test population's lengthy experience reading and identifying Latin characters. Even with training, we cannot reasonably expect our test population to accumulate a comparable amount of experience with a new Smartfont over the course of a study. Our evaluation method does not require training people to read Smartfonts.

3.3.1 Background on Evaluating Fonts

Existing techniques for evaluating text readability and legibility are not readily applicable to Smartfonts. These evaluation methods rely on participants' ability to read the script being tested, but reading Smartfonts requires training. Prior studies typically ask participants to read text and then complete a task based on that text. For example, participants might read paragraphs of text in different fonts and then answer basic comprehension tests (e.g. [61]). Reading time and comprehension level serve as metrics for readability. An alternate setup consists of presenting a paragraph of text with individual word substitutions (e.g. [14, 36, 13]). The number of word substitutions detected measures readability or legibility. Other tests involve showing a single word or pseudoword, and asking the person whether the word they saw was a real word (e.g. [42]). Accuracy in distin-

guishing words from non-words in relation to word display time determines legibility. These tests would dramatically favor traditional Latin characters due to the participants' experience in reading them.

A motivating starting point for our Smartfont evaluation techniques is the work on human perception by Pelli et al. [118]. This work compares the "efficiency" of letter identification across traditional and made-up alphabets. Efficiency was measured by how well individual letters could be identified in the presence of random noise, which is different but possibly related to blur. They also found that a few thousand training examples sufficed to teach someone to identify unfamiliar letters fluently. Pelli's methods for evaluating character distinguishability and learnability inform our Smartfont legibility and learability test designs.

3.3.2 Experimental Setup

Our legibility experiments consist of showing participants a target string and asking them to select the matching string from a list, as shown in Figure 3.10. The targets were random strings of length five, roughly the average word length for English. We chose strings as visual stimuli based on the theory that words are preferable to both individual letters and sentences for evaluating visual acuity [11]. Individual letters are inappropriate since they cannot blend with neighboring characters as longer strings do, and longer text is not ideal because individual factors unrelated to visual clarity contribute to reading comprehension (like inference from surrounding words). Each question came with four possible answer choices. One of the answer choices was the same five-letter string as the target. The other three matched four out of five target characters, with one random replacement.

We used a within-subject design, with each participant answering questions for a single Smartfont and for Latin. Latin text was presented in Helvetica. The target image was presented at decreasing sizes, with three questions at each size for both fonts. Blur was manipulated with a Gaussian filter, which replaces each pixel with a weighted average of nearby pixels. A large radius creates highly distorted images and mimics severe presbyopia, while a small radius leaves images largely intact. The blur radius was fixed throughout each experiment. We first scaled text then applied blur. This simulates the experience of reading small, blurry text (e.g., a presbyope reading

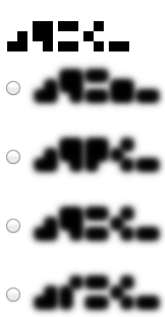


Figure 3.10: Sample legibility task with Smartfont Visibraille.

text on a smartphone), where the stimulus is fixed and clear and the eyes essentially apply a blurry filter.

Simulating blurry vision allowed us to crowdsource our evaluation through Amazon Mechanical Turk.[5] Instead of screening for specific blurry vision conditions, we blurred text and asked the crowd to "read" it. Low visual acuity and other problems early in visual processing are in some sense transformations on the stimulus. Whether blurring occurs on the screen or in the visual system, the perceptual effect is similar. Consequently, there is a precedent for blurring images to simulate blurry vision in reading studies (e.g., [84]).

We ran two experiments. Our first compares all five Smartfonts at a fixed blur of 3.5. We had 154 participants (69 female, 81 male, 4 other). Ages ranged 18-72 (mean 35). 32 evaluated Polkabet; 30 evaluated Visibraille 2; 31 evaluated Visibraille; 33 evaluated Tricolor; and 28 evaluated Logobet. 69 were wearing glasses during the study, and 85 were not. Our second experiment compares Tricolor at three different blurs. It involved 104 participants (37 female, 64 male, 3 other). Ages ranged 20-69 (mean 36). Of these, 36 saw a blur of 2.5; 33 saw a blur of 3.5; and 35 saw a blur of 4.5. Among the participants, 41 wore glasses during the study, and 63 did not. Participants received $1.50. Workers quit and were paid when the size became too small, so few answered all questions. Workers had at least a HIT (Human Intelligence Task) Approval Rate of 97% and 1000

[5] http://www.mturk.com

approved HITs.

3.3.3 Legibility Results

It is not obvious how to compare the legibility of Smartfonts, especially with experiments performed "in the wild" where users have varied screens and software. To address this, we compare, for each participant, the smallest size they can read Latin text to the smallest size they can read their Smartfont. Since each participant viewed Latin and a single smartfont, this enabled us to compare, on an individual-by-individual basis, the Smartfont and Latin letters under the same conditions.

Determining a metric for font size applicable to diverse scripts is challenging. Vision scientists typically use visual angle between the bottom of the text, the viewer's eye, and the top of the text, while typographers prefer the physical print size of characters [88]. Variance in character height and width within fonts further complicates defining size. To fairly compare different scripts, we use *text area* as our metric. Text area includes the white space between and around characters required to render the text that cannot be occupied by surrounding text.

We define a *Minimal Reading Area* (MRA) for font f, MRA_f, which is specific to the participant (and blur). Note that in our experiment we asked three questions for each font at each size. As we decrease size, we say the participant fails to read at the first size where they make a majority of errors (2 out of 3). We define the MRA to be the just-larger size used before failure. Although participants were asked to attempt reading at smaller sizes, this further data was not used in the analysis because it typically reflected random guessing. We also exclude data from participants who failed to read at the largest size, since they likely misunderstood the instructions or were guessing. It is convenient to consider the log-MRA, shown in Figure 3.11, since a constant difference in log-MRA reflects a constant factor change in legible size.

To get some intuition for the meaning of text area, a Facebook post on a Chrome desktop[6] web browser today appears in a font whose full ascender-to-descender height is 13 pixels, which corresponds to a log area 8.75 (see the vertical line in Figure 3.11) in our experiments. With the

[6]Most of our participants were using desktop, not mobile, browsers. See `http://facebook.com` and `http://google.com/chrome`.

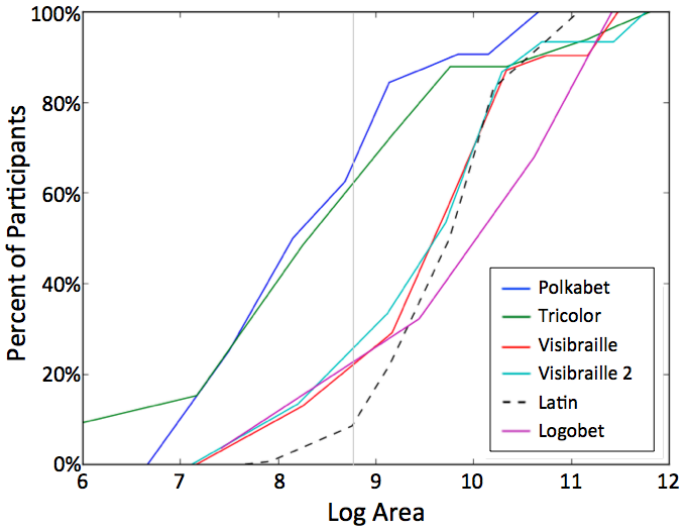


Figure 3.11: The (smoothed) empirical distribution function of log-MRA (at blur of radius 3.5 pixels) for the fonts. y is the fraction of participants whose log-MRA was larger than x.

interpolation in Figure 3.11, this suggests that only 8% of participants could read this size text, at our blur, in the Latin font while over 60% of the participants could read Polkabet and Tricolor fonts. Our definition of MRA focuses on font sizes large enough that users are most often correct, based on reading research which shows that users prefer and read faster at font sizes for which they can readily discern letters (e.g [36]).

To quantify each Smartfont's performance by a single number bounded by a confidence interval, we define the *log-score* (LS), for each experiment to be the log of the ratio of the MRA for Latin to the MRA for each Smartfont f, or equivalently,

$$\mathrm{LS}_{Latin,f} = \lg \frac{\mathrm{MRA}_{Latin}}{\mathrm{MRA}_f} = \lg(\mathrm{MRA}_{Latin}) - \lg(\mathrm{MRA}_f),$$

where lg denotes base 2 logarithm. A log-score of 0 means the participant read the Smartfont at the same size as Latin; 1 means they read the Smartfont at half the size; 2 means they read the Smartfont 4 times smaller; etc. Note that our experiment is inherently one-sided: upper bounds on log-score do not bound the legibility of the Smartfont *after training*.

A histogram of the log-scores for Tricolor is displayed in Figure 3.12. 26 of the 33 participants

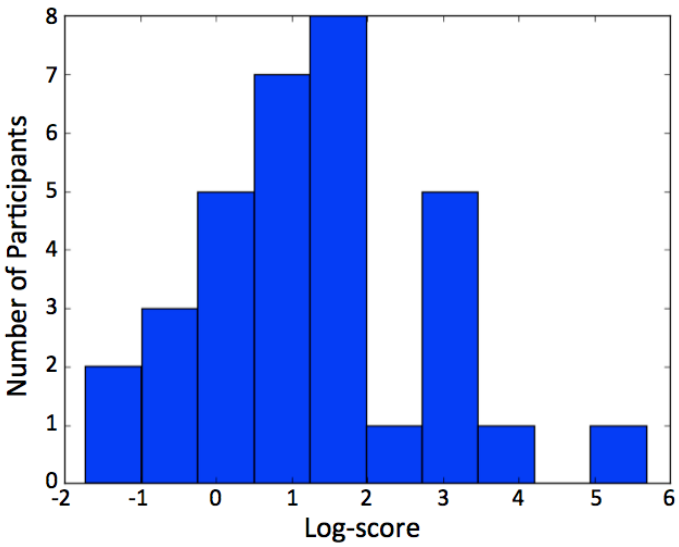


Figure 3.12: Histogram of log-scores for Tricolor with blur radius 3.5 pixels.

(79%) had positive log-scores, meaning that they "read" the Smartfont at a smaller size than Latin, and 18 of the 33 (55%) had log-scores greater than 1, meaning they "read" the Smartfont at least half as small as Latin. The sample mean log-score was 1.28. The wide variance in this histogram means that some users might benefit significantly more than others from adoption.

Table 3.1 displays confidence interval bounds for our Smartfonts. For *simultaneous* 95% post-hoc confidence intervals for five Smartfonts, we choose what would normally be 99% confidence intervals bounding each (the union bound on the 1% failure probability of each estimate then implies 95% confidence). Since our test is inherently one-sided, as mentioned, we use simultaneous one-sided confidence intervals, based on mean and standard deviation. Only Tricolor and Polkabet's confidence intervals are entirely positive, suggesting particularly strong legibility for these Smartfonts.

To see how performance would vary as we change the blur parameter, we compared Tricolor versus Latin at three different blur radii. A larger blur radius corresponds to "blurrier" vision, and a larger mean log-score means that the script was typically more legible compared to Latin. The results at radii 2.5 pixels, 3.5 pixels, and 4.5 pixels, were all greater than 0 with statistical

Smartfont	CI lower-bound	Mean log-score
Polkabet	0.78	1.30
Tricolor	0.62	1.28
Visibraille	-0.23	0.14
Visibraille 2	-0.32	0.14
Logobet	-0.56	-1.03

Table 3.1: Mean and 95% simultaneous (one-sided) confidence interval lower-bounds.

significance, though the differences were not statistically significant. The mean log-scores of 1.17, 1.28, and 1.41, respectively, suggest a possible increasing trend.

3.4 Learnability Evaluation

In order for our Smartfonts to be usable, they must be learnable. To evaluate learnability, we designed an online learning system and tracked partcipants' progess. The learning site assigned each visitor a single Smartfont. It provided a tutorial about the Smartfont, yes/no practice questions in the Smartfont, and flashcards for drilling the meaning of individual characters.

3.4.1 Learning Site Design

The site welcomed visitors with a brief Smartfont tutorial. The tutorial presented 1) the mapping between the Smartfont characters and Latin (i.e. "English") characters, 2) a description of the Smartfont's organization, and 3) examples of words in the Smartfont with their Latin equivalents. The site's welcome page provided the tutorial and a chart of the participant's performance over time for self-tracking. Participants could return to the welcome page at any time.

The site provided short yes/no practice questions to help participants learn their Smartfont. The questions were generated via crowdsourcing, consisting of questions from MindPixel [106] and questions we gathered from Amazon Mechanical Turk workers. We screened the questions for

Figure 3.13: Sample practice question for Smartfont Logobet. (*Do fish wear clothing? Yes/No*)

inappropriate content. In total, we used 2739 questions: 1245 with an answer of "yes" and 1494 with an answer of "no." The questions were generally fun and entertaining. Examples include "Is the moon out at night?" and "Are you a celery?".

Practice questions were displayed with Smartfont characters, as shown in Figure 3.13. After receiving an answer from the user, the site showed the question in Latin characters and gave feedback on correctness. While answering practice questions, participants could reference a cheatsheet of the mapping between Smartftont and Latin characters. The Polkabet cheatsheet also provided mnemonics. To view the cheatsheet, a user could click a link above the question and the cheatsheet would overlay the practice question. This design forced participants to use their memory when answering questions, rather than relying entirely on the cheatsheet to look up each character.

To further help participants memorize their Smartfont, we made flashcards available throughout the study. Each flashcard presented a single Smartfont character, and quizzed the participant on its Latin equivalent. Mistaken characters were repeated until the participant got them right.

3.4.2 Experimental Setup

We recruited 23 people to use our Smartfont learning site through Amazon's Mechanical Turk platform.[7] Each participant was assigned randomly to a single font: 8 to Polkabet, 6 to Tricolor, and 9 to Logobet. Varying numbers for each font are due to participant dropout during the study. Participants chose how long they spent on our site. They typically spent 2-3 hours per day on our site over the course of about a week. Our study workers received $5 for their first 10 questions,

[7] `http://www.mturk.com`

$0.05 per question for the next 3333, and an extra $50 if they completed 3333 by the study end date. Workers had at least a HIT Approval Rate of 97% and 1000 approved HITs.

Participants set up accounts on the learning site so that they could log back in to continue learning and we could track their progress. Participants were compensated for the yes/no practice questions that they answered, but were free to use the flashcards, cheatsheet, or tutorial at any time. We recorded time and accuracy in answering the yes/no questions. One in every 10 yes/no questions was displayed in Latin characters for baseline comparison. We also recorded their use of the cheatsheet and flashcards throughout the study. Participants were free to provide open-ended feedback through a form on the site at any point during the study.

3.4.3 Learnability Results

We evaluated learning in terms of speed and accuracy in reading and answering the practice questions. To evaluate speed, we calculated the ratio of the time it took them to answer each Smartfont practice question to the average time it took them to answer the Latin control questions. A value of 1 means that it took the same time to answer Smartfont questions as Latin ones, a value of 2 means it took twice as long, and so on.

All participants held over 95% accuracy in answering the encoded questions, so they were not guessing. Practice did not increase accuracy. Average accuracies were: Polkabet 97.8%, Logobet 97.3%, Tricolor 98.2%, Latin 98.9%. The difference between each Smartfont and Latin was statistically significant ($p<0.001$, Kruskal-Wallis). Accuracy and response time were weakly correlated ($r=-0.0030$, $p=0.2534$, Pearson).

Figure 3.14 shows the general trends across our Smartfonts. Tricolor exhibits the easiest learning curve, followed by Logobet and then Polkabet. After $2,000$ questions, participants learning Tricolor were reading a median of 2.1 times slower than they did in Latin; participants learning Logobet were 5.2 times slower; and participants learning Polkabet were 6.7 times slower. We ran an unpaired t-test to determine whether the differences in response time across fonts was significant after 2000 questions. We found a statistically significant difference between each pair of fonts: Polkabet and Logobet ($t(8498) = 10.6623, p < 0.0001$), Polkabet and Tricolor

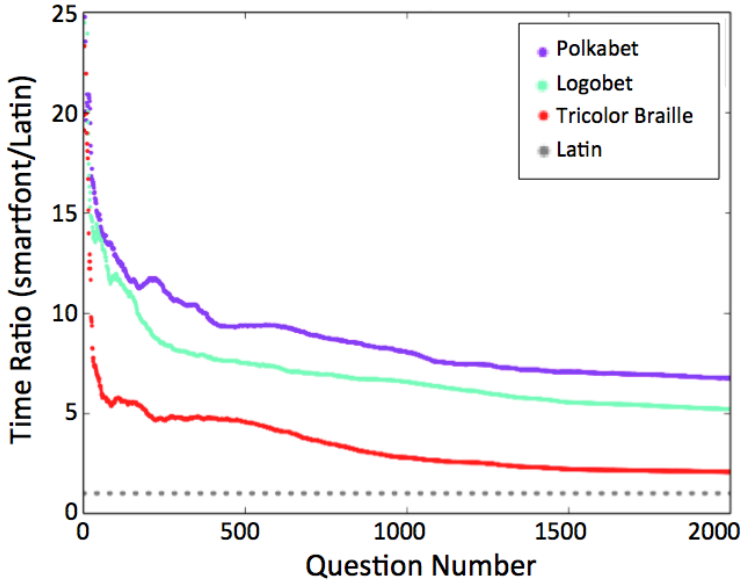


Figure 3.14: Smartfont response time, normalized by individual average Latin time. Each point is the median of a sliding window of averages across participants to remove outliers.

$(t(7998) = 4.0640, p < 0.0001)$, and Logobet and Tricolor $(t(8498) = 2.6588, p < 0.008)$.

There was some variation in learning curves between individual participants. Two participants learned their Smartfonts, Tricolor and Logobet, extremely quickly. They became as fast at reading the Smartfont as they were at Latin after only around $1,000$ questions. Their quick learning curves show that while learning Smartfonts might be somewhat challenging for most people, it is quite natural for some. Those with an affinity for learning Smartfonts have a low barrier to start reading and benefiting from Smartfonts. The variation in learning curves between individuals also suggests the benefit of personalization. Some may prefer to learn Smartfonts that would challenge others, and these preferences may be individual.

Adding color to Tricolor's characters appeared to support learning. Since its characters' shapes are unique and resemble the Latin alphabet, one might be concerned that people learn to read it ignoring color. This did not seem to be the case. We found that participants remembered the colors of common words, not just the letter shapes. In a post-test three days after the site closed,

seven readers of Tricolor were asked to identify the coloring of five common words (like "the") on multiple-choice questions with four choices (three random colorings of the same shaped letters). The aggregate accuracy was 28/35 (80%) over these four questions, strongly indicating that they had remembered at least some of the colors.

Participants used both the flashcard and cheatsheet as they learned our Smartfonts. Participants learning Tricolor made more use of the learning resources than participants learning Polkabet or Logobet. It is possible that participants learning Tricolor relied more on the learning resources because their font tutorial did not include additional information beyond its mapping to Latin characters. We provided mnemonics for Polkabet, which likely helped Polkabet participants recall more characters independently, if slowly. Similarly, we provided a lengthy tutorial for Logobet detailing the character organization, that likely helped participants remember the character representations. Because of its relative simplicity, Tricolor had no mnemonics or details about font organization.

We gave participants the opportunity to provide open-ended feedback. Their responses indicated that they largely enjoyed learning and reading our fonts. The majority remarked that their experience was "fun." Several compared the reading Smartfonts to solving puzzles. One even wrote, "someone should find a way to turn this into an Android game." Participants also noted their progress. One found it, "super hard in the beginning but on the last couple I actually was reading them as though I was seeing the letters." At the end of the experiment, one participant contacted us, asking if they could continue using our site to practice their font. Coupled with our learning curves, the participants' positive reflections suggest that people can enjoyably learn to read Smartfonts fluently.

3.5 Discussion and Future Work

In this chapter, we introduced Smartfonts, scripts that leverage computer grahpics to completely redesign the written alphabet with the purpose of improving the reading experience. We also presented experimental designs for evaluating 1) the legibility and 2) the learnability of Smartfonts under various reading conditions. We do not claim to have created the best Smartfonts or even optimal Smartfonts for reading blurry text, but we have hopefully demonstrated that it is possible

to improve over the millenia-old letters in use today.

There are several limitations to the work presented in this chapter. First, we did not control for screen type, screen resolution, or distance between the viewer and screen, as our experiments were not run in a controlled environment. Moving away from a controlled laboratory setting allowed us to use crowdsourcing for rapid experimentation. However, studying legibility in a lab setting with users with visual impairments, as we do in the next chapter, would be beneficial. Second, we do not currently offer users who learn a Smartfont the ability to use it on their devices, which could be crucial to adoption.

We also limited the design space to create each of our Smartfonts. While the limited design spaces allowed for 26 unique characters, a limited design space makes it difficult to ensure that Latin text is always unambiguously recoverable from Smartfont text for a larger set of characters, for example including both lowercase and capitalized letters, punctuation, and digits. Possible solutions include 1) relaxing design space constraints, 2) using different design spaces for different character sets (e.g. letters vs. digits vs. punctuation), 3) creating compound characters and 4) adding indicator characters at the start of different modes (e.g. text vs. numbers). Braille uses 3) and 4) with a 63-character design space. Color in Smartfonts is especially suited for messaging, where text is typically monocolor, but raises interesting questions in broader graphic design where color serves important functions. We plan to further explore the Smartfont design space, including expanding Smartfonts to characters beyond the alphabet.

We would also like to explore benefits of Smartfonts beyond improved legibility. Smartfonts can provide *privacy* by visually encrypting text.[8] "Substitution ciphers" which encrypt text by replacing each letter with a symbol, have been used by da Vinci in mirror-writing, by Union prisoners in the Civil War, and by children in games and journals. Privacy can be especially valuable on smartwatches, where embarrassing personal communications may appear without warning, visible to anyone sufficiently close. Smartfonts might also affect cost or durability of displays. For instance, the seven-segment digit display common to digital alarm clocks and other electronics is

[8] as long as the particular Smartfont being used is personalized or not widely used

cheaper and has fewer pieces that may fail than a high-resolution screen. Smartfonts could similarly improve printing or hardware costs.

In the future, Smartfonts could even be tailored to an individual's eyesight or display screen. Each person is unique, and a wide variety of vision conditions exist. We imagine a system that evaluates a person's vision and generates optimized Smartfonts on-the-fly. Such a system would require learning a model of how vision relates to script readability. Just as many Southeast Asian scripts have rounded letters because straight lines would tear the palm leaves on which they were written [104], Smartfonts could also be tailored to their display screens.

Smartfonts could also be generalized to other character systems besides Latin. For example, we can develop Smartfonts for the Hebrew alphabet or Chinese characters. Some East Asian scripts are read top-to-bottom, so any Smartfont involving kerning would need to support combining adjacent characters vertically. The size of character sets can also vary enormously. For example, there are over 50,000 Chinese characters. A Smartfont for such a large character set would likely need to take advantage of language or character structure. Like any initial project in a space, Smartfonts introduce many new research directions.

Chapter 4

LIVEFONTS

Can we further improve text legibility by leveraging animation capabilities of computer graphics to differentiate letterforms? How can we design these animated scripts in a principled way?

The previous chapter proposed Smartfonts, alternate scripts that leverage computer graphics to redesign letterforms, and showed that Smartfonts can expand access to textual information by improving legibility. However, the designs presented explored only *some* opportunities computer graphics offer visual text design, namely color, shape, and spacing variations. Computer graphics also support *live, dynamic* displays, and these past designs were stationary. In this chapter, we add animation to the Smartfont design space to create *Livefonts*. To the best of our knowledge, we are the first to propose using animation to differentiate characters in a script.

Intuitively, adding animation to text has the potential to compress text while maintaining legibility. In particular, adding one dimension helps remove certain restrictions on other dimensions that previously limited legibility. For example, 26 characters that vary along a single dimension (e.g., shape) cannot be very dissimilar, and thus the smallest legible size is limited. If the 26 characters may vary along two dimensions (e.g., shape and color), pairs of characters can be more similar in one dimension as long as they vary in the other. Hence, we hypothesize that adding animation to the Smartfont design space will allow us to generate Smartfonts that are legible at significantly smaller sizes than strictly static text, or equivalently significantly more legible at the same size. It is less clear whether or not people would be able to read such animated text.

In this chapter, we present a Livefont in two variations, Version 1 and Version 2, to improve legibility by means of recognition for both low-vision and sighted readers. The designs are informed by iterative design and a perceptual study we ran on animation and color. Unlike the previous

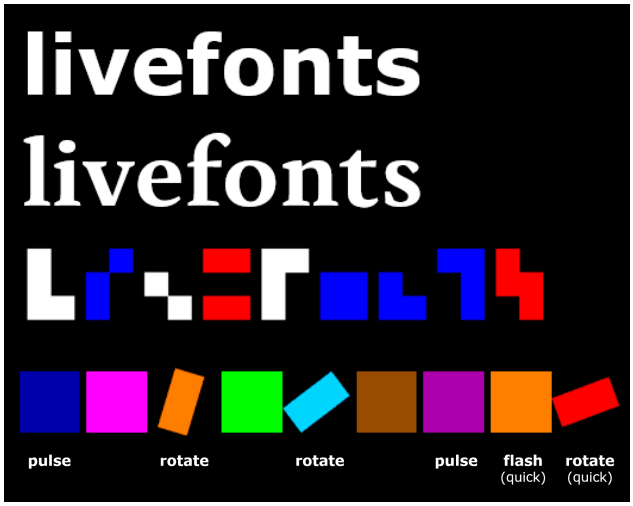


Figure 4.1: The word "livefonts" rendered in: a standard font (Verdana, top), a font designed for low vision readers (Matilda, second), a Smartfont from the previous chapter, and our Livefont (bottom). Note: figures should appear animated when this document is viewed with Adobe Reader.

chapter's remote evaluation of Smartfont legibility using blurred text to simulate low vision, this chapter presents a controlled laboratory study of our Livefont's legibility with both low-vision and sighted participants. We find that Livefonts are legible at approximately half the size of traditional Latin fonts across vision types, even more legible than the Smartfont designs introduced in the previous chapter. While the previous chapter also reported a nearly two-fold area decrease in blurred text for the best Smartfont, this chapter's experiments do not reflect a significant advantage for that Smartfont among low-vision users, which is consistent with previous work on the inadequacy of disability simulation. Perhaps more surprisingly, Livefonts were found to be readable with practice for many people, in an evaluation of learnability similar to that of the previous chapter. This chapter's learnability study involves a new teaching method that eases people into a new script, by providing overlays of traditional letterforms that are gradually removed over time.

Key contributions of this chapter are: 1) the idea of Livefonts, animated scripts to enhance reading, and specifically to improve legibility for low vision; 2) a controlled in-lab study on Live-

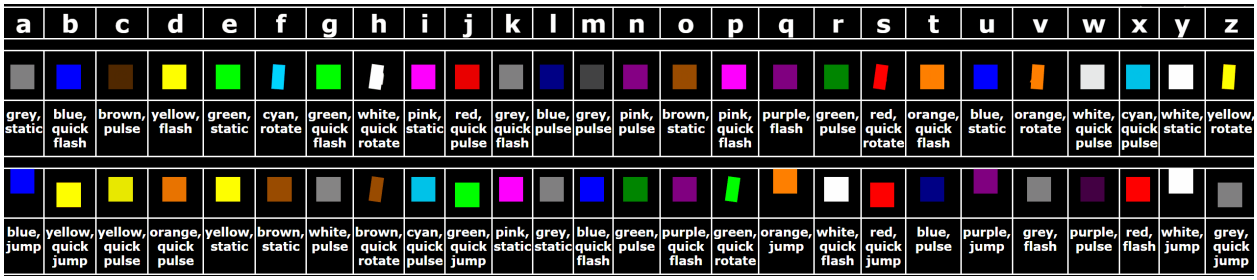


Figure 4.2: Our Livefont versions (top: *Version 1*, bottom: *Version 2*), with descriptions of character color and animation. Characters are colored, animated blocks, chosen to be maximally distinguishable by our color/animation perception study.

font legibility with low-vision and sighted participants, using a novel transcription methodology; and 3) an exploration of Livefont learnability for low-vision and sighted readers.

4.1 Livefont Designs

We present our Livefont in two variations (Figure 4.2). Since color is more easily recognized in large solid blocks than in detailed strokes, its letters are (animated) squares, which maximize character area. To design these Livefonts, we first engaged low-vision readers in an iterative design process to constrict our design space, and then ran a crowdsourced study to fully explore that design space and find the most perceptually distinguishable character sets.

While animated text has the potential to distract or annoy the reader, various techniques can be used to address these potential effects. For example, eye tracking could be used to selectively animate text being read; animations can be triggered by dragging a finger along text displayed on a touchscreen; and the animations themselves can be designed to please the user. In this chapter, we tailored our designs to our target low-vision users based on their feedback. The subsequent chapter (Chapter 5) explores alternate display modes for animated text.

4.1.1 Narrowing the Design Space to Color and Animation

We employed a user-centered design process with low-vision readers to narrow our designs to animated blocks of colors. Because the design space is virtually unconstrained by modern screens, it was important to limit our design space. We chose to involve low-vision readers to best design a Livefont that met their needs, as they are a target group who can potentially benefit enormously from improved legibility. We met regularly with local low-vision people, and remotely with a low-vision visual artist, to solicit feedback on designs. At the initial meetings, we conducted informal interviews to learn about their vision and reading. At subsequent meetings, we showed them designs, and gathered feedback and suggestions. Designs explored included sets of colored dots, abstract shapes, various moving gradients, animated/colored traditional letterforms, and traditional letterforms tailored to low-vision. We found that a black background was generally preferable, and that large blocks of color with simple animations were typically easier to perceive, which became our design space.

Our final color palette was: red, orange, yellow, green, cyan, blue, purple, pink, white, grey, and brown. The colors were hand-selected with input from people with low vision, to be discernible and to support clarity on a variety of monitors and personal device screens. They are spaced out in hue and have distinct English names. Our final set of animations were: static, flash, pulse, jump, and rotate, each (besides static) available at two speeds. Pulse is a gradual increase and decrease in color; flash intermittently shows and hides the block's color; jump is an up-and-down shift in position; and rotate is a clockwise rotation. All characters were squares, except those used for rotations were rectangles, so that the rotations would be visible. All animations run continuously, and were implemented with CSS animations. These animations involve large area changes over time, which we found to be most discernible during the iterative design process. Our design space thus consisted of 11 colors and 9 animations, yielding 99 possible characters and $\binom{99}{26} > 10^{23}$ possible character sets. Further exploration of the color/animation design space is left for future work.

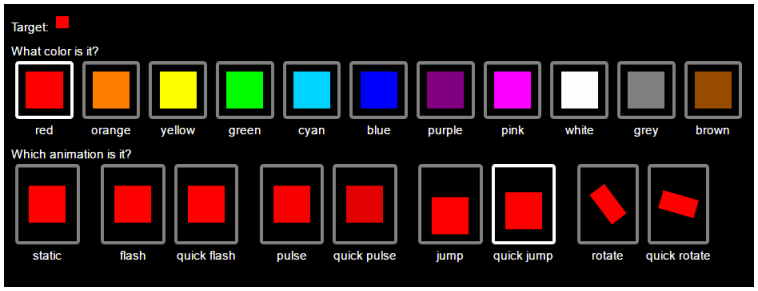


Figure 4.3: Animation/color perception study task. A target block is shown. The participant is asked to identify the target color (red selected) and animation (quick jump selected).

4.1.2 Selecting Alphabet Characters

After narrowing our design space to animated blocks of color, we ran a perception study on the identifiability of characters in our design space at small sizes. This allowed us to choose letters likely to be highly legible. We crowdsourced the study with sighted participants in order to gather sufficient data on the large design space. The study presented a series of animated, colored blocks, and asked participants to identify their color and animation, as shown in Figure 4.3. Target blocks were presented one at a time. Each participant answered 9 practice questions. They then answered 99 test questions, covering all color/animation combinations.

The practice target height was 1em (at 14 point), and the test target height was .15em. We wanted to make targets small enough that they were challenging, to gather data at the limits of perception. Because it was a crowdsourced web survey, it was impossible to control for absolute size or visual angle, but the size was commensurate with typical browser font size. In addition, participants were instructed not to zoom in.

We posted the task on Amazon's Mechanical Turk crowdsourcing platform and recruited 50 participants (19 female, 31 male). Ages ranged 24-69 (mean 35.4). Twenty-six owned glasses or contacts, all but three of whom wore them during the study. Three identified as having low vision; nineteen identified as nearsighted; and one as farsighted. Two reported being unsure if they were

(a) Colors

Red	94.44%
Blue	92.87%
Green	88.20%
Cyan	86.89%
Yellow	83.71%
White	76.39%
Orange	75.67%
Purple	75.28%
Grey	71.49%
Pink	63.31%
Brown	48.66%

(b) Animations

Static	97.99%
Jump	96.72%
Quick Flash	89.78%
Pulse	87.80%
Quick Jump	69.22%
Flash	69.03%
Quick Pulse	56.04%
Rotate	50.46%
Quick Rotate	47.64%

Table 4.1: Color and animation identification accuracy.

colorblind, and the remaining forty-eight identified as not colorblind. No participants identified as having a learning disability or as being dyslexic. All participants except one, who dropped out, evaluated all 99 color/animation pairs.

In total, we collected 5386 evaluations, 49-50 for each color/animation combination (49 evaluations for 14 combinations due to our one drop-out). Individual accuracies in identifying the correct color/animation combination ranged from 0.23 to 0.90 (avg. 0.60, dev. 0.18), perhaps due to variance in visual acuity. Mean accuracies in identifying colors and animations are shown in Table 4.1. Red and blue were the most accurately identified colors, and brown was the least. Out of our animations, static and jump were identified most correctly, with the two rotation speeds least accurately identified.

To obtain our final Livefont design from this data, we adopted the optimization procedure from the previous chapter. Specifically, we created a 99 x 99 confusion matrix, with rows and columns

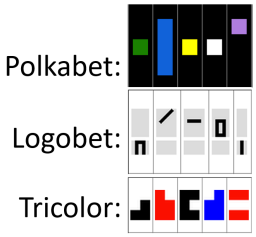


Figure 4.4: The static Smartfonts (letters A-E) to which we compared Livefont Version 1.

representing all animation/color combinations. We then select the 26 rows (and corresponding columns) that yield the lowest net confusion. Specifically, let $c_{i,j}$ represent the fraction of times character i was confused for character j. Then we choose the set of 26 characters S that minimizes $\sum_{i,j \in S, i \neq j} c_{ij}$. Because this is an NP-hard problem (see Appendix A.1 for a proof), we used a branch-and-bound algorithm guaranteed to find an optimal solution, which terminated quickly. This approach does not consider English letter frequency, and we defer deeper explorations into language-dependent optimization to future work. We performed this optimization twice, once including jumping animations, and once without, to produce two variations. We excluded jumping from one to explore if the additional vertical area required for jumping paid off in increased legibility.

4.2 Legibility Study

To evaluate the legibility of our Livefont, we conducted a controlled laboratory study with both low-vision and sighted participants, unlike the previous chapter's evaluation using remote sighted participants and simulated blurry vision. Because low-vision readers struggle with legibility, meaning letter and word recognition, our study involves letter and word recognition tests. It does not evaluate readability in terms of comprehension. Future studies are needed to study longitudinal usage, and effects of animating long passages.

The experimental design is within-subjects across a range of traditional fonts and Smartfonts (sampled in Figure 4.4). The study had one session, divided into two parts: 1) a transcription task

	Owns Glasses		Wore Glasses		Nearsighted			Farsighted			Colorblind		
	Yes	No	Yes	No	Yes	No	IDK	Yes	No	IDK	Yes	No	IDK
Low-vision (10)	9	1	8	2	6	2	2	3	7	0	2	7	1
Sighted (15)	13	2	11	4	11	4	0	11	4	0	0	15	0

Table 4.2: Self-reported vision descriptions from our legibility study participants, separated into low-vision and sighted groups. Due to variability between low-vision users and even between an individual's two eyes, it can be difficult to answer these questions. IDK stands for "I don't know."

measuring script acuity ($\sim$ 55 min), and 2) a scanning task measuring visual scanning time ($\sim$ 5 min). Participants were compensated $20.

We recruited 25 participants (10 low vision, 15 sighted). Participants varied in age (15-67, mean 34), and gender (15 female, 10 male). Sighted participants were recruited through relevant email lists from the local population. Low-vision participants were recruited from local low-vision mailing lists and support groups. To verify that our participants had low vision, we conducted a brief screening interview. Our low-vision participants had a range of vision conditions including ocular albinism, retinitis pigmentosa, nystagmus, retinopathy of prematurity, and Norrie disease, resulting in a range of reading challenges, in particular difficulty with small letters. Because low vision is very diverse, we did not further categorize low-vision participants by condition, though further improvements may be possible by addressing conditions separately. Participant responses to high-level vision questions are shown in Table 4.2.

All study procedures were completed using a computer with a standard monitor. All scripts were rendered with a black background. For all scripts other than Smartfonts we used white, bold versions, to yield the best results for low vision, and the black characters of Tricolor, the static Smartfont we used for comparison were made white for visibility on the black background.

4.2.1 Part I: Transcription Methodology

We employed a novel evaluation methodology based on transcribing characters at increasingly small sizes. Evaluating the legibility of Smartfonts, scripts that nobody knows how to read, is difficult. Methods for testing acuity typically involve identifying letters by name (e.g., an optometrist's Snellen chart), or reading. These methods do not apply to Smartfonts without extensive training. In the previous chapter's evaluation, participants identified 5-character strings using multiple-choice options that differ by a single character, at increasingly small sizes. However, this method produces a single piece of information with every task, namely the single mistaken character. Our transcription methodology provides more data with every task, namely which characters were misread as which others.[1]

Because visual acuity varies greatly, especially among low vision, we first calibrated text size for each participant. We presented a list of random[2] sentences in a traditional font, at increasing sizes, and asked participants to select the "smallest readable size," as done with a MNREAD acuity chart [102]. A chin rest was used throughout the study to fix the distance from the screen and control angular text size.

After calibration, participants completed a series of transcription tasks. The task, shown in Figure 4.5, presented a target string of five randomly chosen characters. Participants transcribed the target characters in order, using an on-screen keyboard. As they clicked on matching characters, their partial guess appeared below the target. The keyboard for the two Latin fonts and for the Latin-esque Tricolor Smartfont adopt the standard QWERTY layout. Livefont keyboards were organized into animation-by-color matrices. Rows were organized by animation, and columns by color. Absent characters were left blank. This design supported visual search by color or animation, helping to even the comparison to transcribing traditional letters with the familiar QWERTY

[1] To see the informational advantage, suppose all letters were clear except *m* was easily confused for *n*. In order for this to be discovered in a multiple-choice test, a pair of words would have to be generated which differed by an *m* replaced with a *n* (or vice-versa), which happens on less than 5% of randomly chosen questions, whereas 32% of random 5-character strings used in transcription tasks would have an *m* or *n*, each of which is an opportunity to identify the confusion.

[2] Random sentences from the random sentence generator `http://www.randomwordgenerator.com/sentence.php/` (accessed Aug. 2016)

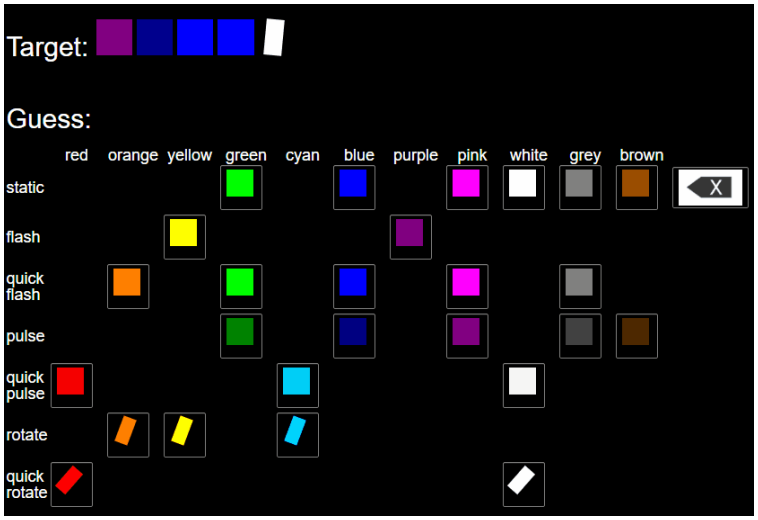


Figure 4.5: Transcription task with Livefont Version 1 – a target string (and partial guess), with a visual keyboard of Livefont Version 1 characters for transcription. (See Figure 4.2 for the mapping of Version 1 characters to the Latin A-Z.)

keyboard. All keyboards contained a backspace button for corrections.

Each participant completed transcription tasks for all scripts, randomly ordered. Participants transcribed targets from each script at decreasing sizes until failure, when they proceeded to the next script. Each script began at 1.5 times the calibration size, to provide practice before reaching a size where where mistakes were likely due to limited acuity. Each subsequent target was 90% the size (area) of the previous. We operationalized area by normalizing each script's height to yield the same alphabet area, computed as the area of the smallest enclosing rectangle. We stopped participants when they made at least 6 mistakes across two trials to prevent participant frustration and data collection on random guesses. The scripts evaluated were: a traditional font (Verdana), a font specifically designed for low-vision reading (Matilda), our Livefont variations, and the "best" static Smartfont from previous work (Tricolor).

Figure 4.6: Scanning task with Latin (Verdana Bold). The target string (top) is a random 5-character string. The selected matching string in the random pseudo-sentence is outlined in white (bottom).

4.2.2 Part II: Scanning Methodology

After transcription, participants completed scanning tasks. The task presented a random 5-string target, which participants identified in a random pseudo-sentence (Figure 4.6). The sentence contained 10 strings, one of which was identical to the target. The other 9 strings were generated randomly, with length between 1 and 8. A limitation of this design is that string length can be used as a cue during scanning. They familiarized themselves with each target before viewing the sentence. The time between the sentence's appearance and when they clicked on the match was recorded internally. Selected strings were outlined in white. Corrections could be made by deselecting and selecting a new string. When satisfied, they clicked "Done", and were shown the correct response. Each participant completed five scanning tasks per script.

Script order was randomized. The scripts used were: Verdana, Matilda, Tricolor Braille, Version 2, Version 1, Hebrew, Arabic, Armenian, Devangari, and Chinese. These scripts were chosen for diversity, and taken from previous work [118]. To control for variance in alphabet size, we chose 26 random lowercase characters to represent scripts with more than 26.

4.2.3 Legibility Study Results

Evaluating our legibility study results requires controlling for variance in script size and eyesight. To compare scripts that vary in character height and width, we use alphabet area as the metric of size, as in the previous chapter. The metrics from the previous chapter do not apply here, as they are task-dependent, and we are evaluating legibility through a new study design. To compare

individuals with varied acuity, we normalize individual results for each script by their results for traditional letterforms (Verdana). This yields normalized metrics for both transcription (the *Area Ratio*) and scanning (the *Time Ratio*). Using these metrics, we find evidence that our Livefonts are legible much smaller than traditional letterforms, and might support faster scanning with practice. However, due to small sample size and noise, follow-up studies with larger populations are needed to confirm our results.

4.2.4 Part I: Transcription Results

To quantify how small people could make out each script, we define a metric called the *Area Ratio*. As described above, each participant reached a smallest legible size for each script, defined as the first size where they failed to transcribe at least 6 out of 10 characters for that script. To account for differences in acuity across participants, we normalize this failure size with respect to the participant's Latin failure size. We call this ratio their *Area Ratio* for a particular script. The Area Ratio is 1 for any participant with Latin. A value lower than 1 means that the script was more "legible" than Latin, and a value above 1 means that it was less legible. For example, a score of 0.5 means that that script was legible at half the size (area) of Latin, for that participant. We note that an n-fold reduction in area corresponds only to a $\sqrt{n}$-fold reduction in font size according to more standard one-dimensional metrics.

The Area Ratios for our participants are shown in Figure 4.7.[3] For both low-vision and sighted participants, Version 1 was generally legible at the smallest sizes, at approximately half the size of Latin, with a minority of participants reaching sizes 4-6 times smaller than Latin. We ran one-way ANOVAs with repeated measures and found statistical significance between scripts for both sighted $(F(3,14) = 13.59 p << 0.05)$ and low-vision $(F(3,9) = 3.19, p = 0.04)$ groups. Post-hoc paired t-tests with Bonferroni correction show statistical significance $(p < .0083)$ for sighted transcription between Version 1/Matilda and Tricolor/Matilda. These results suggest that Livefonts can improve legibility for both low-vision and sighted readers, though follow-up studies are needed

[3] In all box plots, the red line is the median. The box lower and upper limits are the 1st and 3rd quartiles. Whiskers extend to 1.5 IQR (interquartile range) in either direction from the 1st and 3rd quartiles.

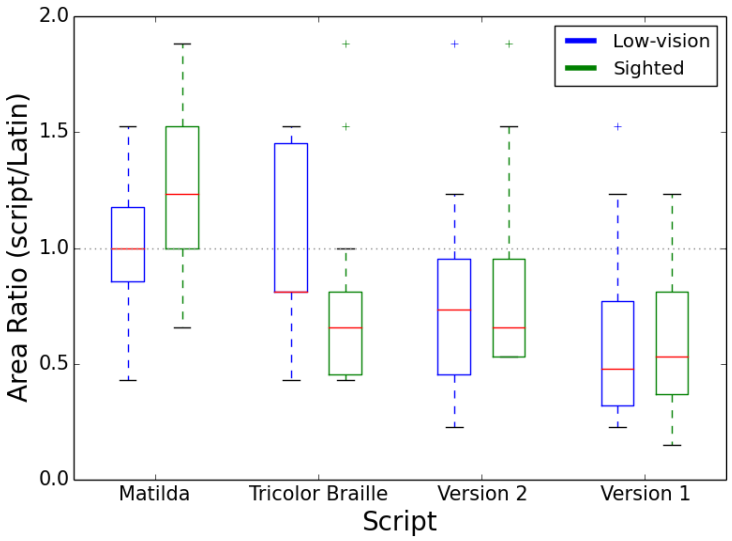


Figure 4.7: Transcription results. Box-plots of participants' smallest legible size for that script, normalized by their smallest legible Latin size. Lower means more legible.

for verification.

We also examined the breakdown of transcription errors by color and animation. The average accuracy in identifying color and animation – Version 1: color 74%, animation 74%; Version 2: color 72%, animation 73% – suggests both were salient identifying features. The error distribution for Version 1 is shown in Figure 4.8. Among colors (Figure 4.8a), blue characters were most often mistaken for other blue characters. This coincides with participant feedback during the study that the blue characters were hard to see on the black background. Green, white, brown, and grey were also commonly mistaken for other characters of the same color. White characters were often transcribed in place of a variety of colors, perhaps due to the neutrality of white making it a natural random guess. Transcription of white characters for red is due to Version 1 containing both a white and red quick rotate. Among animations (Figure 4.8b), quick rotate and quick flash were often mistaken for static characters. It is likely that as size decreased, the rotation was lost. Pulse was commonly guessed in place of a variety of animations, possibly due to its sharing properties

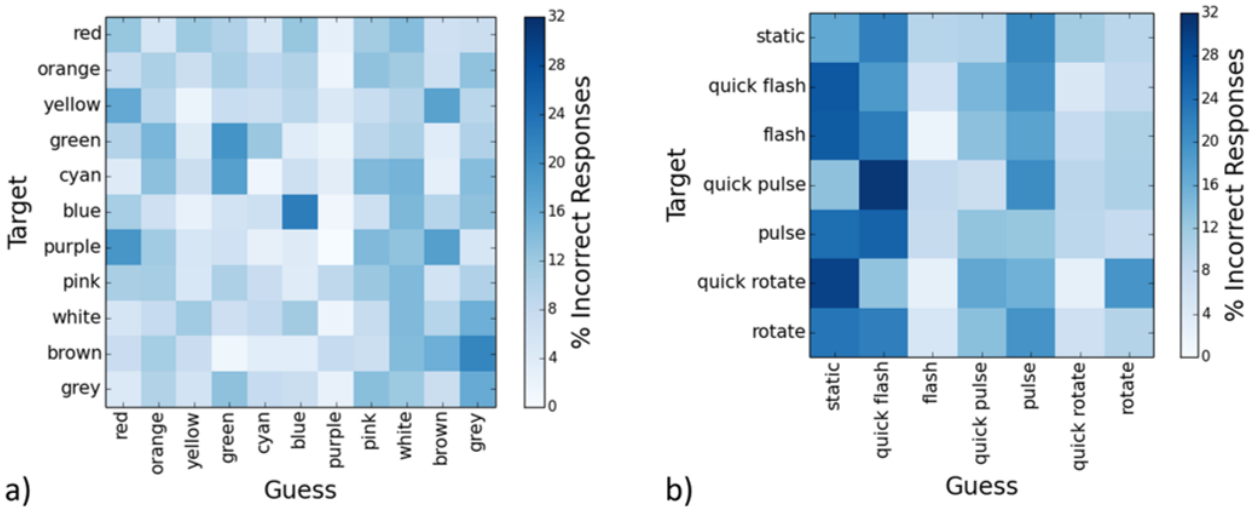


Figure 4.8: Distribution of errors in identifying a) colors and b) animations of Livefont Version 1 across all participants in the transcription task. Each row shows the breakdown of mistakes in identifying that color or animation. (See Figure 4.2 for Version 1 alphabet.)

with many animations (e.g., a similar on-off pattern to quick pulse, flash, and static flash). Similar trends exist for low-vision and sighted groups separately, with sighted errors generally more evenly distributed. Some participants also reported visual fatigue and expressed annoyance at some characters, in particular the blinking ones, while others described the task and scripts as fun.

4.2.5 Part II: Scanning Results

To evaluate our scanning results, we define another normalized metric, the *Time Ratio*. For each participant and every script, we compute a *Time Ratio*, defined as their median scanning time for that font divided by their median Latin scanning time. We normalize by Latin scanning time to account for innate variance in scanning speed. Time Ratios for each script are plotted in Figure 4.9. We also ran one-way ANOVAs with repeated measures to evaluate statistical significance.

Our Livefonts yielded relatively fast scanning times, compared to traditional unfamiliar scripts. Matilda generally produced the fastest scanning times, likely because Matilda uses Latin letter-

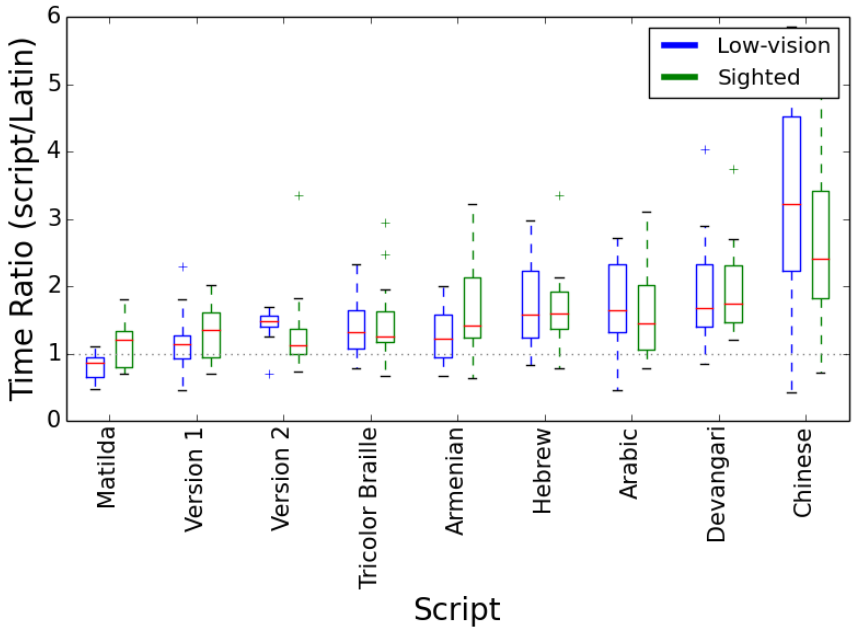


Figure 4.9: Scanning results. Box-plots of participants' median scanning time finding a 5-character string, normalized by their median time with Latin. Lower means faster.

forms, which are easier to identify due to familiarity. Furthermore, those letterforms are tailored to low vision, which likely boosted scanning speed for low-vision participants. For both low-vision and sighted participants, Chinese produced the slowest scanning times, likely due to the absence of additional spacing between Chinese characters in adjacent words and complex Chinese character design. One-way ANOVAs with repeated measures reveal statistical significance between scripts for both sighted $(F(8,14) = 7.39, p << 0.05)$ and low-vision $(F(8,9) = 8.96, p << 0.05)$ groups. Post-hoc paired t-tests with a Bonferroni correction show significance $(p < .0014)$ for sighted scanning between Matilda/Devangari, Matilda/Chinese, Version 1/Chinese, Version 2/Hebrew, Version 2/Devangari; and for low-vision between Version 2/Matilda. Our Livefonts' comparable scanning times to the fastest foreign scripts suggest they might yield faster scanning times with practice, though follow-up work is needed to fully explore potential scanning benefits.

4.3 Learnability Study

As noted in the previous chapter, for a new character system to be useful, it must be learnable. To evaluate our Livefonts' learnability, we adopted the previous chapter's evaluation design, where participants learn to read Smartfonts through encoded practice questions online. Unlike their evaluation with only sighted people, we recruit both low-vision and sighted participants.

4.3.1 Learnability Study Methodology

We recruited 15 participants (7 sighted and 6 low-vision) for our learning study through Amazon Mechanical Turk. Recruiting low-vision participants was a two-step process. First, we ran a survey on the platform to gather information on people's vision, without any hint of future work. The survey consisted of 10 questions to probe whether or not they were low-vision. This included questions on how they identified (typically sighted, blind, or low-vision), what vision conditions they have been diagnosed with if any, what visual aids they use, and whether their vision is correctable with glasses. 543 people responded, 12 of whom we identified as low-vision. Second, we advertised our learning study to these 12. Sighted participants were recruited from the general Mechanical Turk population by offering 12 workers direct access to our learning study. Our survey showed that only about 2% of workers are low-vision, so the probability of obtaining low-vision workers in the general recruitment is very small.

During the study, participants visited a website that taught them to read Version 1. We chose to study Version 1 over Version 2 due to its better performance in our legibility study. The site teaches the user the new script through several components: 1) an introductory tutorial explaining the Livefont structure and providing the alphabet 2) encoded yes/no questions, and 3) flashcards to drill individual character meanings. We used the same 2739 crowdsourced questions as in the previous chapter, a supplemented set from MindPixel [106].

The yes/no questions, pictured in Figure 4.10, were the primary teaching tool, and response time was the primary metric we used to evaluate learning. A cheatsheet was available upon demand during the yes/no practice questions, showing the alphabet and including mnemonics we de-

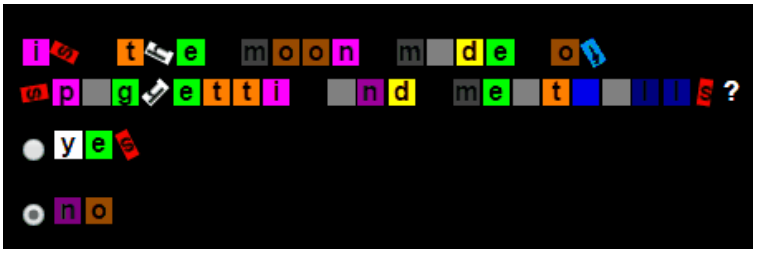


Figure 4.10: Sample yes/no practice question (*Is the moon made of spaghetti and meatballs? Yes/No*). Here, some letters are overlaid with Latin letterforms to ease the learning curve.

signed to help memorability. The cheatsheet overlaid the current question, forcing the participant to remember what they learned from the cheatsheet in order to answer the question. Every tenth question was not encoded (in plain English, using Latin letterforms), for a control comparison.

To ease learning, we initially overlaid Livefont characters with their traditional Latin representations, and gradually removed the overlays. At the start of the study, all characters were overlaid with Latin. Every 45 encoded questions, another letter's overlay was removed, in alphabetical order, so that after 1170 encoded questions (1300 total questions), no characters were overlaid with Latin, and participants were forced to rely entirely on their memory, plus the supplemental learning aids. This differs from the learnability experiment in the previous chapter, where learning was upfront based upon rote memorization and mnemonics, and this difference should be taken into consideration when comparing results across studies.

Participants were paid $5 for the first 10 questions. After that, they were paid on a per-question basis. They were not paid directly for their flashcard use, though flashcard drills could improve their hourly rate by improving their speed. If they reached the end of the study, they received a $50 bonus. Because low-vision reading is typically slower than sighted reading, we paid low-vision participants 7 cents per yes/no question, and sighted participants 5 cents per yes/no question. The site was in operation for 10 days.

Several days after the learning study closed, we distributed a survey to obtain feedback and

gauge how much learning had converted to longer-lasting memory. The survey quizzed participants on the animation and color of randomly chosen letters, asked participants to rate usefulness of site resources, and gathered open-ended feedback. Participants were paid $5.

4.3.2 Learnability Study Results

Our primary metrics of learning are time spent answering the yes/no questions, and accuracy. Reading time is a preferred metric in psychophiscs research [87], and accuracy reflects content understanding.

Learning Accuracy

All participants maintained a high level of accuracy through the experiment. Average (mean) accuracy was 98.23% (min 97.59%, *SD*=0.50%) among low-vision participants, and 97.79% (min 96.67%, *SD*=0.75%) among sighted participants. Given that with random guessing the expected accuracy would be 50%, it is safe to assume that participants were processing question content. The difference between each group's Livefont and Latin accuracy was statistically significant ($p<0.001$, Kruskal-Wallis). Accuracy and response time were significantly correlated for our low-vision group (r=-0.0226, p=0.0062, Pearson), but not for our sighted group (r=0.0027, p=0.7842, Pearson). Interestingly, the correlation for low-vision participants is opposite what would naively be expected – increased time is associated with a *decreased* accuracy (or vice versa). It is possible that while time might help the eyes focus and gather more information, additional time is predominately indicative of difficulty or frustration.

Learning Speed

The fast initial reading speed and subsequent slowdown for Version 1 for both low-vision and sighted participants, as shown in Figure 4.11a, is attributable to the overlaid Latin letters we initially provided. The learning curves for both low-vision and sighted participants peak well before all letters are hidden, at 1170 questions. It is likely that providing overlays for the letters at the end

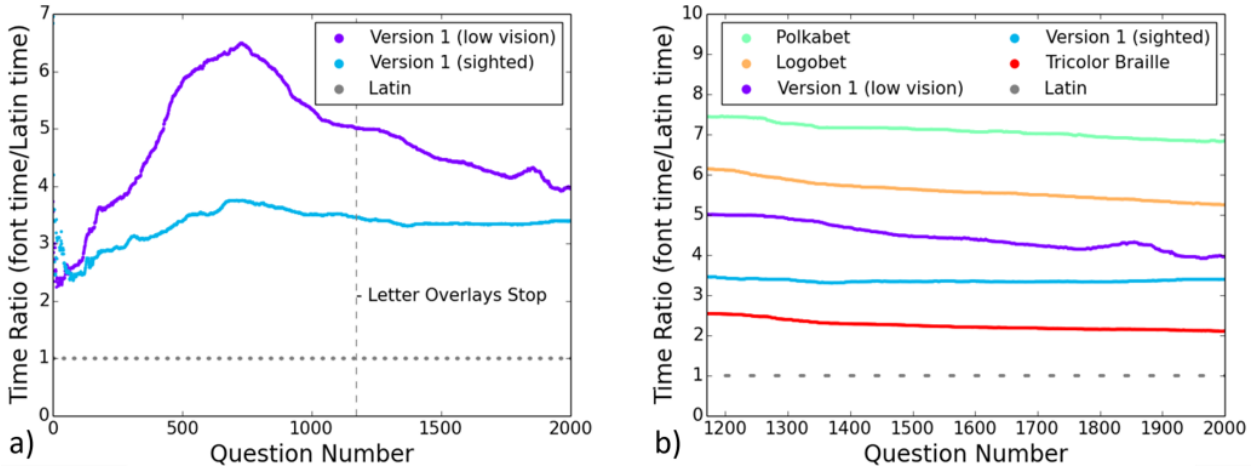


Figure 4.11: Average Livefont Version 1 response times, a) the full learning curve for Version 1, with an initial letter overlay aid, and b) the tail of the Version 1 learning curves compared to static Smartfonts. Results are normalized by individual average Latin time. Each point is the median of a sliding window of averages across participants to remove outliers.

of the alphabet (e.g., *x*, *y*, and *z*) did not have an impact because these letters are rare, especially in simple sentences like those used in the experiment. The fast initial speed, and low peaks of about 3.5 and 6.5, compared to average starting times of up to 25 times slower than Latin in the previous chapter, suggests that overlaying Latin letters can significantly reduce the effort required to start learning a Smartfont. It could lower the barrier to learning Smartfonts, and make them a more practical option for more people. However, a comparison between these two learning methodologies on the same font would be necessary to verify this conjecture.

The learning curve was initially steeper for low-vision participants, compared to sighted ones. It is possible that this difference is due to an increased effect of removing the overlaid letters for low-vision participants. Removing the overlaid letters forced participants to gradually rely on their color/animation perception alone, which might have been an easier transition for sighted participants. Nonetheless, the normalized speed of our low-vision readers approaches that of the sighted

participants as they approached 2000 practice questions.[4] The difference after 2000 questions was not statistically significant, according to an unpaired t-test ($t(6498) = -1.0221, p = 0.3067$). If low-vision and sighted participants continue this trend past 2000 questions, the average low-vision participant might reach or surpass the average sighted participant.

We also compare Version 1 learnability to stationary (non-animated) Smartfonts (Figure 4.11b). The stationary Smartfonts (Figure 4.4) were produced by the same experimental setup as in the previous chapter. However, in that experiment, no Latin overlays were provided, so we start the comparison where our overlays finished. As shown, reading speed with our Livefont is comparable to other Smartfonts after 2000 practice sentences. Reading speed for both low-vision and sighted participants was faster than all Smartfonts except Tricolor. We ran unpaired t-tests to determine statistical significance at 2000 questions, and found statistical significance between low-vision Version 1 times and each static Smartfont.[5] Normalized response times for the last 500 questions were our measures for each participant. No statistical significance was found between sighted Version 1 and static Smartfont response times.[6] Note that the static Smartfont results were produced with sighted participants. It is possible that the significant difference for low vision is due to this difference in vision rather than Smartfont design.

Individual learnability of Version 1 varied greatly among both participant groups, as shown in Figure 4.12. Among low-vision participants (Figure 4.12a), the Livefont was particularly learnable for P1 and P4. These two participants almost reach their Latin reading speed with 2000 practice sentences. On the other hand, Version 1was not very learnable for some participants, in particular P3, who made very little improvement in reading speed and whose Version 1 speed was over 5 times slower than Latin. A similarly wide range in learning is exhibited by our sighted participants (Figure 4.12b). The previous chapter also reported a large variance between participants in

[4]In particular, it is possible that low-vision participants used strong magnification, which forced letter-by-letter reading. Guessing the meaning of a letter that newly lost its overlay mid-word could be very difficult. In contrast, a sighted participant might more easily guess a newly missing overlay given more context.

[5]Polkabet: $(t(6998) = -17.00, p < 0.0001)$, Tricolor: $(t(5998) = 17.94, p < 0.0001)$, Logobet: $(t(7498) = -13.60, p < 0.0001)$.

[6]Polkabet: $(t(7498) = 0.03, p = 0.9739)$, Tricolor: $(t(6498) = 1.66, p = 0.0964)$, Logobet: $(t(7998) = 0.60, p = 0.4874)$

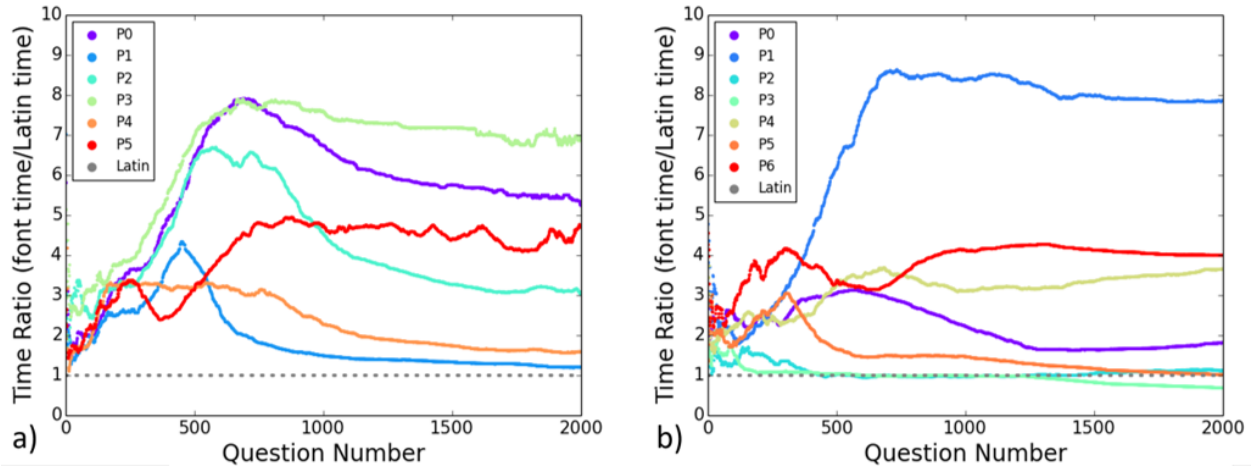


Figure 4.12: Individual learning curves for a) low-vision and b) sighted participants, for Livefont Version 1. Results are normalized by individual average Latin time. Each point is the median of a sliding window, for smoothness.

their learning experiment. While it is difficult to attribute this learnability disparity to visual or cognitive differences, the spread suggests the need for personalized Smartfonts or a wider range of Smartfonts from which to choose. Alternatively, the potential Smartfont user population might not include everyone.

Post-Study Survey

Four low-vision participants and five sighted participants completed the survey we distributed after the learning study closed. Up to a week after practicing, they identified character color with 58% accuracy[7] and animation with 46% accuracy.[8] Given that with random guessing we would expect 9% color and 14% animation accuracy, it seems that participants did commit characters to memory. Interestingly, they remembered more character colors than animations, though there were more colors than animations to confuse. A number of possible explanations could account for this: our

[7] 64% accuracy for low-vision, 53% accuracy for sighted participants

[8] 36% accuracy for low-vision, 54% accuracy for sighted participants

mnemonics were more helpful for colors, our colors were more memorable than our animations or had more memorable names, or colors are simply easier to remember than animations.

Participants' average evaluation of resource usefulness (on a scale of 1-5), in order, were: 1) overlaid letters (4.6, std. 0.7); 2) tutorial (4.3, std. 1.1); 3) cheatsheet (3.9, std. 1.1); 4) flashcards (3.7, std. 1.5). They typically found the overlaid letters most helpful. One participant summarized the benefit, "It gave me a lot of help by learning to read the words step by step. By omitting characters this way, it's less of a shock than them disappearing all of a sudden." Participants also found the study generally fun and stimulating. As one participant concluded, "I liked progressing. That was fun."

4.4 Discussion and Future Work

Livefonts offer exciting possibilities of improved legibility for (small) screen devices, especially for low-vision readers. Magnification helps low-vision readers distinguish letters, but the accompanying loss of visual context and required panning are inconvenient at best. Increased legibility from Livefonts can potentially help reduce or eliminate the magnification needed to identify letters. Sighted users can also benefit, especially people reading small text on small screens, those who wear glasses but do not always have them at hand, and people who need glasses but cannot afford them.

While we present the first animated scripts, this work has several limitations. First, we do not claim to have created an optimal animated script. There is a virtually unlimited design space for Livefonts, and we only examine two possibilities in this space. Our experiments also have limitations. They do not evaluate readability comprehensively, but rather legibility in terms of character and word identification, and learnability in terms of understanding short sentences. While letter and word identification are fundamental to reading, we do not measure the legibility of long excerpts of text. We also have not studied the long-term impact of reading animated Smartfonts. Given the small sample sizes of our studies, larger studies with diverse users are needed to confirm our results and better understand the research space.

Livefont design is a rich space for future work. The use of color and animation can potentially

distract or annoy the user and inhibit reading, in particular for color- or motion-sensitive readers. Ideally, users would choose from Livefonts with varying color and motion patterns to best suit their sensitivities. For practical considerations, the present work focused on designing and evaluating two options. Long-term studies, beyond this work's scope, are needed to understand and design Livefonts to mitigate these effects. Interactions between adjacent animations or colors can also impair or aid legibility. A thorough understanding of such effects, combined with data on bigram and trigram frequencies, would make it possible to replace English characters with animated characters so as to minimize undesirable neighbor effects. Selectively animating traditional Latin letterforms could also improve text legibility without requiring readers to learn a full set of new characters.

The effects of spacing and timing on Livefonts also offer rich opportunities for study. Animations can be sped up or slowed down, and it would be interesting to study which speeds best suit which types of vision, and to see how many distinct speeds of a single animation can be distinguished – in this work we only use two. Synchronization across characters can also yield powerful effects. For example, characters blinking in unison create a unifying effect across the page, whereas staggering can help blinking characters blend in. A "wave" effect can also be made by slightly offsetting adjacent letters, which could be used to boost reading speed by guiding the eyes through the text. Perhaps most compellingly, users can be given control over animations. For example, readers could choose between viewing modes with different animation settings, or animation speed could adjust automatically to the participant's reading speed detected through eye-tracking. This initial work on character systems defined by animation introduces many research questions to a variety of fields including design, typography, psychophysics, human-computer interaction, and accessibility.

Chapter 5

ANIMATED SI5S

Can animating an ASL character system help remove barriers to adoption and use?
How can we design these animations in a structured way with input from ASL users?

Text-based resources present information access problems for ASL users, because text is not generally available in ASL. Sign languages lack a standard written form, preventing millions of Deaf people from accessing text in their primary language. A standard written form of ASL would make text-based platforms newly usable in ASL, including email clients, text messengers, social media, text editors, and much of the internet, as well as text-based resources like books and newspapers. It would also satisfy many Deaf people's desire to express themselves through writing in their primary language (e.g., [149]). Other possible benefits include promotion and dissemination of the language, and low-cost documentation and cultural preservation [59, 105].

A major barrier to ASL character system adoption and use is difficulty learning a system which represents complex 3D movements with stationary symbols. As noted in the previous chapter, character systems worldwide have historically been defined by stationary (non-moving) features. Our paper medium, which only supports stationary displays, has limited character design in this way. Representing the 3D movements of ASL in stationary 2D notation is difficult. As a result, the extent to which a stationary character system can resemble live signing is limited, making it difficult to learn and use.

Unlike paper, computer graphics support animation, providing an opportunity to create character systems that more naturally represent sign languages (as well as character systems that are more legible, as explored in the previous chapters). Incorporating animation in text is particularly relevant to sign languages like ASL, which are movement-based. In particular, unlike stationary symbols, animation can easily indicate gradation in sign speed, which has semantic meaning in

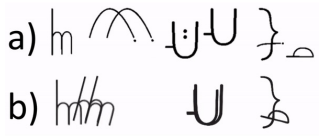


Figure 5.1: YOU GO TO SCHOOL TOMORROW. in a) si5s, and b) our Animated Si5s prototype. The movement symbols in a) (the arcs and dots) are replaced by animating the referenced handshape symbols to create b). (This figure should be animated when viewed in Adobe Reader.)

ASL. Furthermore, we increasingly read material on computerized devices, so character designs no longer need to be limited to stationary characters.

In this chapter, we leverage the animation capabilities of computer graphics to present Animated Si5s, the first animated sign language character system prototype. Like Chinese characters and heiroglyphic logograms, our animated characters represent individual signs, and form lines of text (e.g., Figure 5.1). Each character is composed of a configuration of symbols, some of which may be moving. These characters are based on si5s [10], formed by replacing si5s movement symbols with actual movement on the screen. Using animation allows us to create text that visually resembles sign movements. This visual resemblance to the live language has the potential to make character systems easier to learn and lower the barrier to adoption.

Key contributions of this chapter are: 1) Animated Si5s, an animated character system prototype for ASL, derived from an existing stationary character system, si5s, 2) a survey and pilot study pointing to the need for an animated character system for ASL, and 3) identification of design dimensions for representing sign movements in a character system, and guidelines for appropriate designs based on a participatory design workshop with ASL users.

5.1 Opportunity Evaluation Study

To better understand potential animated ASL character system users, we ran an online study involving three parts: 1) a survey on ASL character systems, 2) a pilot study on whether animation

can make a character system easier to understand without training, and 3) feedback on animated vs. stationary character systems. It was designed to answer four main questions:

Q1. Do ASL community members want to read content in an ASL character system?

Q2. If so, what is preventing them from using ASL character systems?

Q3. Can animation make an ASL character system easier to understand without training?

Q4. Does the community of ASL users see value in creating an animated ASL character system?

5.1.1 Procedure

The study was run online as a web study, and took 10-15 minutes total. After obtaining consent and asking basic demographic questions, the study consisted of three main parts:

Part 1: Survey questions on how people communicate in ASL on traditionally text-based platforms, and their usage of ASL character systems. The questions were multiple-choice, and allowed multiple selections and write-in options as needed.

Part 2: Pilot study on animated ASL character identifiability, probing whether animation can improve notation understandability without training. Participants were asked to identify signs from notation without training (Figure 5.2). Participants viewed notation for each sign in both animated and stationary forms, in order of increasing complexity. We randomized whether participants saw the animated or stationary set first. After viewing each set, participants provided qualitative feedback on how easy the signs were to identify, how enjoyable viewing them was, and how similar they looked to live signs.

Part 3: Feedback on the usefulness of animated ASL characters. We explicitly asked participants if they thought animating ASL characters can be valuable, and why (or why not).

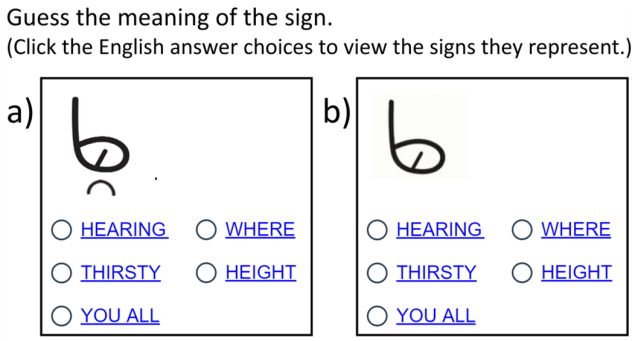


Figure 5.2: Sample identification questions for the sign WHERE, from text in a) si5s and b) Animated Si5s. (Viewed in Adobe Reader, b) contains animation.)

5.1.2 Identification Questions

We based the pilot study (Part 2) on si5s because it naturally integrates animation (see Section 5.2 Animation Design for details). The stationary notation of each sign was presented in si5s. The animated version was created by replacing si5s movement symbols with animations. Specifically, the handshape symbols were animated according to the drawn trajectory, direction, or pattern. The animations were designed by the researchers based on experience with typography, animation, and ASL. The animations were created in PowerPoint, presented as GIFs, and played continuously.

The identification questions involved four signs: WHERE[1], UNDERSTAND[2], MAYBE[3], and MOTIVATION[4], chosen to span different movement types and complexities. Each question provided five multiple-choice answer options, ordered randomly. Answer choices were selected to resemble the represented sign by matching features that characterize the sign (e.g., handshape, body location, etc.) against a database of features for $> 1,000$ signs.[5] Each multiple-choice option was represented as an English word, with a link to a signed video from SigningSavvy [128], an online English-to-ASL dictionary.

---

[1] `https://www.signingsavvy.com/sign/WHERE/478/1`

[2] `https://www.signingsavvy.com/sign/UNDERSTAND/715/1`

[3] `https://www.signingsavvy.com/sign/MAYBE/261/1`

[4] `https://www.signingsavvy.com/sign/MOTIVATION/3924/1`

[5] Source and details anonymized for review purposes.

5.1.3 Participants

ASL users were recruited online, through relevant email lists and social media, with IRB approval. In total, 195 participants completed the survey (74% completion rate). ASL proficiency was self-reported, along the 6-point IRL Scale [125]. Basic demographics: **Age**: 9–77 (m=35); **Gender**: 147 (75%) Female, 47 Male, 1 Other; **Identity**: 98 (50%) Deaf, 18 (9%) Hard-of-Hearing, 74 (38%) Hearing, 5 (3%) Other; **ASL Level**:

Level	# Participants
0 - No proficiency	0
1 - Elementary proficiency	21 (11%)
2 - Limited working proficiency	19 (10%)
3 - Professional working proficiency	64 (33%)
4 - Full professional proficiency	46 (24%)
5 - Native or bilingual proficiency	45 (23%)

5.1.4 Results

The results from our opportunity evaluation study show a desire for an easily understood ASL character system, interest in an animated character system, and potential for animation to make character systems more immediately understandable.

Q1: Need for ASL Character Systems

The vast majority of participants reported a need or desire for an ASL character system. When explicitly asked which materials they want available in ASL text (Figure 5.3), the vast majority – 86% Deaf/Hard-of-hearing (DHH), 71% hearing participants – selected at least one material type, indicating a widespread desire for access to text in ASL.

Participants' means of communicating in ASL (Figure 5.4) further support the potential value of an animated ASL character system. A high percentage reported using some form of digital ASL communication (>95% DHH, 82% hearing), suggesting a pervasive desire for digital ASL

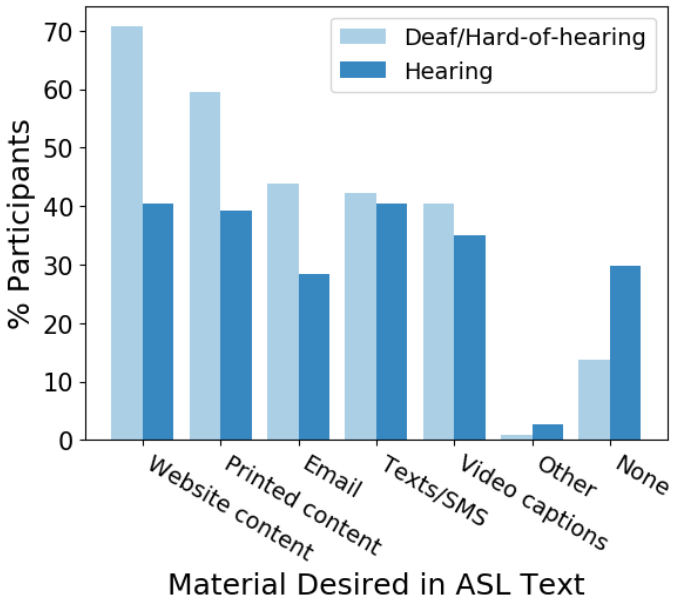


Figure 5.3: Materials participants reported wanting to read in ASL text.

communication. Despite the availability of video, text-based methods were the most common among DHH (animated emoji, and English gloss), suggesting a need for ASL text, in addition to video. The popularity of animated emoji (55% DHH, 14% hearing) suggests that integrating animation into text-based platforms is highly desirable, in particular for DHH people, which an animated character system would provide.

The vast majority of participants (91%) reported taking notes for ASL content (Figure 5.4b), further supporting a need for reading/writing in ASL. Example use cases include preparing a vlog post, taking lecture notes, or writing a note or poem in ASL. The most-reported note-taking methods were text- (not video-) based, likely due to the ease, low cost, and inconspicuousness with which text can be created and edited.

Most participants (>90%) reported being able to read at least one ASL writing system (Figure 5.4c), indicating a wide need for them. English gloss was more popular among hearing participants, who are typically native English speakers using gloss to learn ASL. In contrast, ASL character systems were more popular among DHH than hearing participants, a difference underscoring the

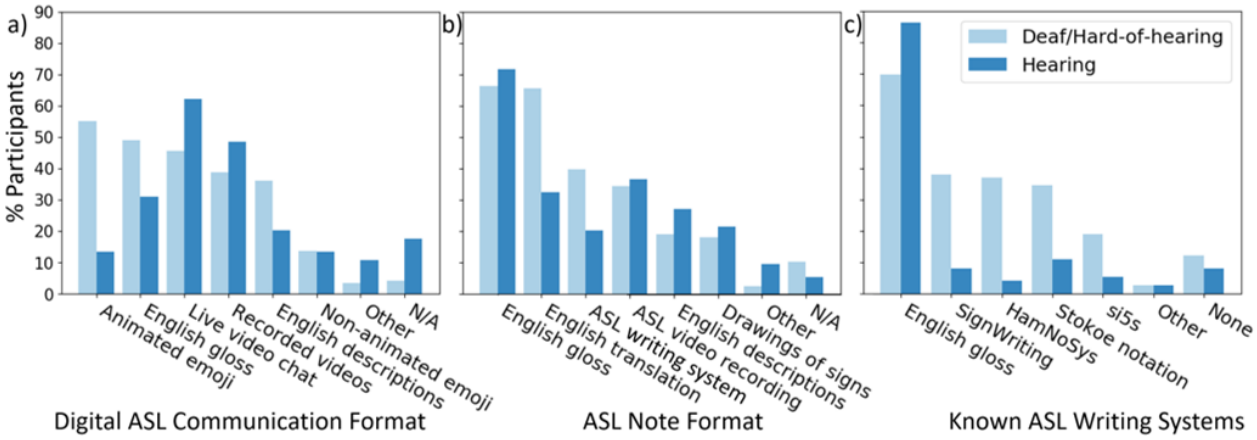


Figure 5.4: How people communicate in ASL a) digitally, b) when taking notes, and c) when using an ASL writing system.

Deaf community's need for an ASL character system not based on English. We also note that our survey, and recruitment, was in English, excluding Deaf people who do not know English from participating, who we expect to be a significant group based on prior literacy studies. Consequently, the fraction of DHH participants who report knowing English gloss is likely higher than for the DHH community at large.

Q2: Unmet Needs with Existing ASL Character Systems

Behind lack of materials, difficulty learning character systems was the most-reported adoption barrier for DHH participants, preventing adoption for 43% (Figure 5.5). Poor resemblance to ASL was the most common barrier for hearing participants, preventing adoption for >50% of them. An animated character system is designed to address both these barriers.

Q3: Animated Character Understandability

Figure 5.6 shows Identification Accuracy, the percent of participants who identified the sign from its notation correctly. The identifiability of the animated characters was significantly higher for

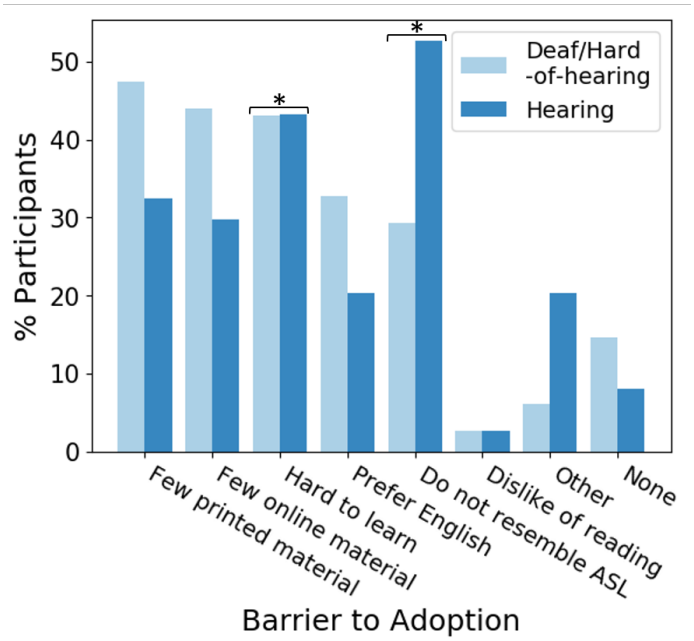


Figure 5.5: Barriers to using ASL character systems reported by participants. Barriers with asterisks are potentially addressed by introducing animation to character systems.

three of four signs (WHERE, UNDERSTAND, and MAYBE).[6] We suspect that animating the character for MOTIVATION did not improve identifiability due to poor design choices, as the animation speed more closely resembled the speed of other signs in the answer choices. The difference between groups who saw animated vs. stationary versions first was not significant ($p > .05$), by a chi-squared independence test for each question. These results suggest that animation can improve character identifiability, and highlight the importance of design choices in creating animations that resemble signs.

Participants took significantly less time to identify signs from the animated notation, suggesting that the animated versions were typically easier to recognize. Even participants who viewed the animated notation first were typically faster at identifying animated characters than stationary ones. The difference in time distributions for identifying the animated vs. stationary version of each sign

[6]Note that the probability of randomly guessing correctly for a single question is .2. It follows that the probability of achieving $>= 60\%$ accuracy (e.g., animated character for WHERE) at random $< .000001$.

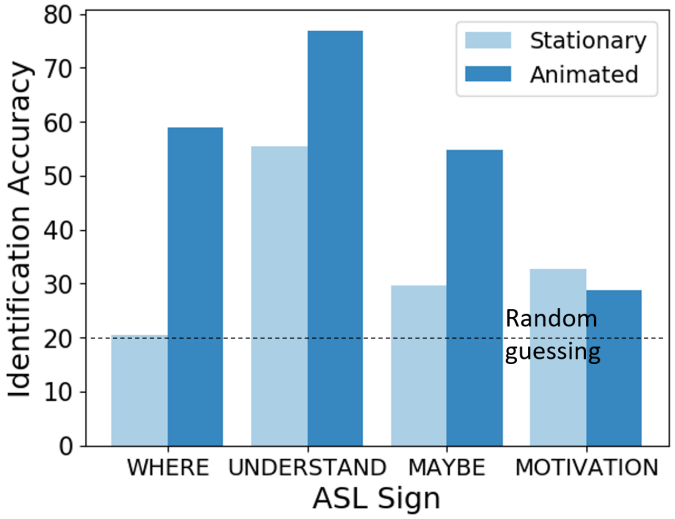


Figure 5.6: Identification Accuracy, the percent who identified signs from stationary vs. animated characters without training.

was statistically significant ($p < .05$), except for WHERE ($p = .16$), by t-tests.

Q4: Qualitative Feedback

Our animated characters received higher ratings than the stationary ones in terms of 1) similarity to live ASL, 2) ease of identification, and 3) viewing enjoyment. Figure 5.7 shows participants' assessments of similarity, ease, and enjoyment on a 7-point Likert scale. For the stationary characters, over 50% of participants reported negative assessment along the three dimensions, compared to under 40% for the animated versions. Overall, the distribution for each answer is skewed towards the positive for the animated characters, suggesting that animation can improve character set resemblance to live ASL, lower the learning barrier, and increase enjoyment.

The vast majority of participants found value in animating ASL characters, as shown in Table 5.1. When asked at the end of the study, "Do you think that animating ASL characters can be valuable?", 71% responded Yes, 22% I'm not sure, and only 7% No. The overwhelmingly positive response to this question strongly suggests that animated ASL characters have potential value to

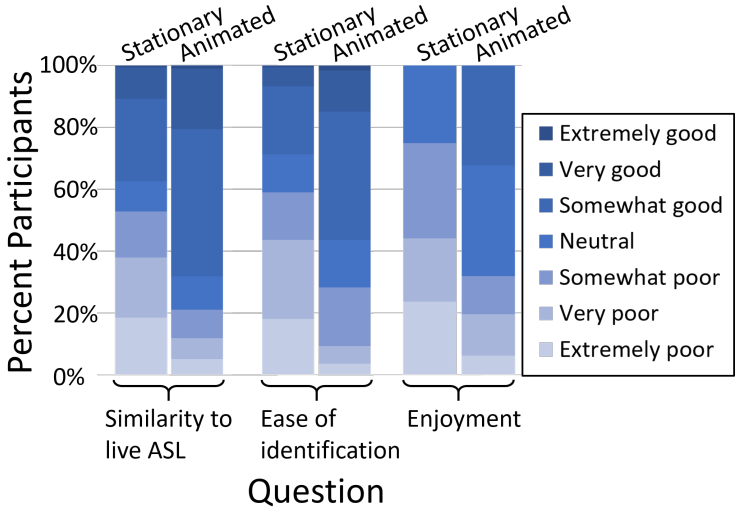


Figure 5.7: Participants' feedback on the animated vs. stationary characters in terms of similarity to live ASL, ease of identification, and enjoyment. Darker means better.

the community, and merit further exploration.

When asked "Why or why not?" after "Do you think that animating ASL characters can be valuable?" the 13 participants who answered "No" rejected ASL writing systems in general, not animation specifically. Some thought ASL is not suited to being written, for example explaining "ASL is not a written language and trying to fit it into that niche feels wrong." and "This is not what ASL is all about - it should be video based only." Others thought that learning a writing system was too hard, one participant explaining, "Too much work... Easier to type in English", and another "It is like learning a new ABC. We have to learn what each symbol means."

5.2 Animation Design

How to design effective animations for a sign language character system is an open question that this work introduces. Our guiding research question through the design process was: What design dimensions (or decisions) need to be addressed, to create an animated ASL character system? To answer that question, we formulated design goals and derived design dimensions from those goals,

Answer	% Participants	# Participants
Yes	71%	139
I'm not sure	22%	43
No	7%	13

Table 5.1: Responses to "Do you think that animating ASL characters can be valuable?"

described as follows.

5.2.1 Framework

We limited the design problem to animating an existing character system, si5s [10], in order to build on substantial prior work and include influences from the Deaf community. We chose si5s for the following reasons:

- It was designed by Deaf community members, and endorsed by Gallaudet University (e.g., [156]), the leading university in the world serving primarily Deaf students.

- It is featural, meaning that it represents sign features separately with component symbols, which allows for movement symbols to be replaced by animations of other symbols (as in Figure 5.2). For example, where the notation indicates that a hand moves up and down, we remove the motion symbol and animate the referenced handshape symbol instead.

- Its handshape and location symbols are fairly iconic, meaning that they roughly resemble handshapes and body parts, which contributes to understandability without training.

- Elegance and simplicity of the system.

- The symbols comprising the character set are publicly available for download from ASLized [8].

5.2.2 Design Goals

Our goal was to animate the line-drawn symbols of si5s, so that the animations resemble live sign movements. Based on related work, knowledge of ASL, and our experience with both character systems and animation design, we broke this high-level goal down into specific goals:

1. **3D movement:** The animations should represent 3D movements (left/right, up/down, and in/out).

2. **Hand orientation:** The animations should represent changes in hand orientation, which occur in 3D space.

3. **Speed:** The speed of the sign movement, which can communicate meaning, should be represented.

4. **Start and end:** The animations should indicate where movements start and end, including whether the movement is repeated, and if so how many times.

5. **Availability:** The animations should be available (present on the screen) when the reader wants them.

5.2.3 Design Dimensions

In order to design effective animations, we broke the design problem down into a series of decisions. To ensure that the hand orientations considered are comprehensive and linguistically sound, they are taken from Stokoe notation [140].[7] We created at least two designs for each design dimension. The design dimensions are outlined below, with our designs presented in Table 5.2.

1. **X-axis movement:** Movement left/right relative to the signer.

[7]The only hand orientation pair omitted – straight or bent wrist – is not explicitly written in si5s, which does not depict the wrist.

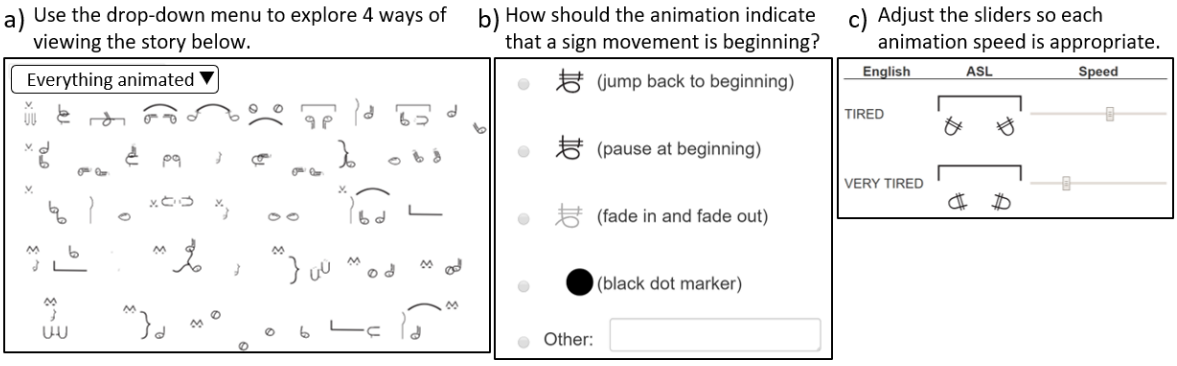


Figure 5.8: Screen shots of the workshop website.

2. **Y-axis movement:** Movement up/down relative to the signer.

3. **Z-axis movement:** Movement in/out of the signer's body.

4. **Facing up or down:** The directions that the fingers point.

5. **Toward or away from signer:** Whether the palm faces toward or away from the signer.

6. **Toward or away from center:** Whether the palm faces toward the center of the body, or away.

7. **Palm up or down:** Whether the palm faces up or down.

8. **Start of movement:** In what position the movement starts (which can be ambiguous since animations play on a loop).

9. **Repeated movement:** How many times a movement is repeated in the sign.

10. **Display mode:** Which portion of text is animated and displayed, and how to navigate the text.

5.3 Design Probe Workshop

To understand which designs ASL users prefer for each design dimension (described above), and to determine if the system is usable in practice, we ran a design probe workshop with deaf and hard-of-hearing ASL users. We used participatory design principles to involve potential users in the design process.

5.3.1 Participants

The workshop was run at a university serving primarily Deaf students, with IRB approval. Participants were recruited through relevant email lists, fliers, and word-of-mouth. A total of 15 students participated. Group demographics: **Age**: 20-41 (m=28); **Gender**: 10 Male, 5 Female; **Identity**: 13 Deaf, 3 Hard-of-hearing; **ASL Experience**: all ASL users, 8 from childhood, 7 from young adulthood; **Known ASL character systems**: 8 SignWriting, 3 English gloss, 1 si5s.

5.3.2 Procedure

The workshop took place in a university conference room, for 1.5 hours. Interpreters were available throughout. In compensation, participants received snacks, drinks, and a $20 Amazon gift card. The basic procedures were:

1. Introductions by everyone present. (5 min)

2. Slide presentation by the researchers explaining the idea of creating an animated ASL character system. (10 min)

3. Participants visited a website to view designs for each design dimension, and input their preferences. (15 min)

4. Participants implemented animated characters by hand in groups of 2-3, for a given set of signs. (30 min)

5. Open discussion. (30 min)

5.3.3 Materials

Slides The introductory slide presentation consisted of slides explaining the concept of designing an animated character system, and providing description and background on si5s.

Website The participants accessed a website using their personal computers (laptops). The website 1) collected basic demographics, 2) showed participants our designs for each design dimension and asked them to select their preference, and 3) asked for open feedback.

Participants selected their preferred design for each design dimension through multiple-choice questions, with write-in "other" options (Figure 5.8a-b). The animated designs were implemented in Javascript/HTML. In-line tables were used to build characters comprising multiple symbols. A long passage [37] was implemented for each display mode (dim 10), with line wrapping (a standard feature of text on computers).

Participants were asked to tune the animation speed for three pairs of signs using a slider (Figure 5.8c). The signs were: TIRED/VERY TIRED, STROLL/WALK FAST, and HAPPY/VERY HAPPY. These pairs of signs were chosen to involve the same movement trajectory, differentiated only by execution speed. The website provided a set of sliders for tuning. When a slider was dragged, the speed of the corresponding animation updated accordingly. The animations were implemented as CSS keyframe animations, with Javascript linking the slider to the display.

Drawing Materials Participants were given printed paper packets, providing instructions for drawing character animations for each of four signs, blank cartoon strips for their drawings, and space to write descriptions and thoughts. They were given pencils, and plastic stencils of si5s symbols at various sizes created by the researchers using 3D laser printers. The signs were those chosen for the opportunity evaluation study to represent diverse sign movements: WHERE, UNDERSTAND, MAYBE, and MOTIVATION (Figure 5.9).

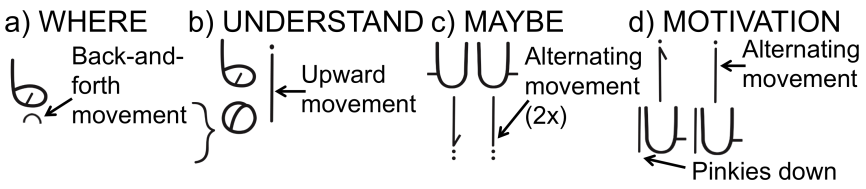


Figure 5.9: The packet's four signs, in si5s. Participants replaced movement symbols (annotated) with drawn animations.

5.3.4 Results

The workshop revealed a preference for designs that visually resemble live signing, showed that the system was viable in that people could implement (create content in) it, and highlighted subjective decisions required to create animations.

Design Dimension Preferences

Participants' selected preferences for the design dimensions are shown in Table 5.2. For representing 3D movements – movement direction (dim 1-3) and changes in hand orientation (dim 4-7) – participants typically preferred designs that mimic how these aspects of 3D movements are perceived in live signing. For movement direction (dim 1-3), participants preferred horizontal translation, vertical translation, and size change, respectively (mimicking a hand moving left/right, up/down, and towards/away the face, respectively). For changes in hand orientation (dim 4-7), participants preferred rotations and reflections, resembling a hand rotating or flipping; and horizontal stretching, resembling an angled hand with one part closer (larger) than the rest. Participants largely agreed on these designs, with >50% agreement for each dimension.

Participants had a lower level of agreement for how to represent a movement's start, number of repetitions, and display mode (dim 8-10). To represent the start of a movement, the most popular design (at 5/15 participants) was no signal – not using any marker to indicate the movement's start. Because animations play on a loop, in this design the movement start is ambiguous, suggesting that some level of ambiguity in exchange for simplicity is acceptable, as in other writing systems.

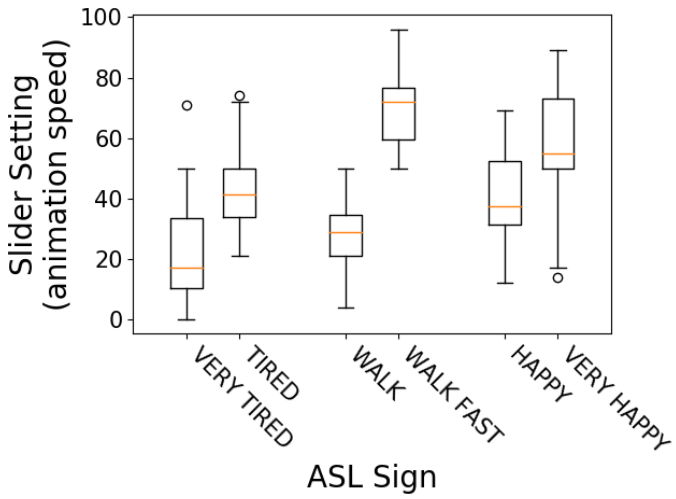


Figure 5.10: Participant animation speed selections. Slider range (0-100) corresponds to a linear time scale for a single animation repetition (5-0.4s). Higher is faster.

For representing an N-time repetition, the most popular designs were an arrow with the N written and a set of N dots above the animation (both at 5/15 participants), suggesting that both are viable options. For display mode, most participants (8/15) preferred the single sign view, though a significant number preferred the on-demand and full page animated designs, suggesting that giving users a choice of display modes is appropriate. In the future, even more display modes could be offered, e.g., using eye tracking to trigger animations.

Animation Speed Preferences

Participants systematically and consistently tuned animation speed, suggesting that animation speed is a salient design dimension, useful for differentiating ASL characters. Figure 5.10 shows the speeds that participants selected for the animated notation of six signs. Significantly different speeds were selected for pairs of signs that differ only by speed in live signing – TIRED/VERY TIRED ($t(14) = 3.02, p = .005$), WALK/WALK FAST ($t(14) = -8.66, p << .005$), HAPPY/VERY HAPPY ($t(14) = 2.10, p = .046$).

Figure 5.11: One group's animation design for the sign WHERE.

The community's consensus around relative speeds suggests that 1) animation speed can affect perceived meaning, and 2) the community generally agrees on what speeds are appropriate for various sign characters, suggesting it is possible to create an animated character systems with speeds that the community will largely accept. Furthermore, while speed is difficult to represent in stationary notation (and is not represented in the stationary si5s notation), speed is an inextricable property of animation. This natural portrayal of a semantically meaningful aspect of the language highlights the potential benefit of an animated character system for ASL.

The System in Practice

All seven groups successfully implemented the system (designed animations to replace si5s movement symbols, e.g., Figure 5.11) for at least one sign. Six groups drew animations for all four signs. The remaining group only animated WHERE. We believe they were unsure how to draw animations for the more complex signs. Given that participants received only a 5-minute introduction to the character system and had limited drawing time, it is likely that with practice or further instruction, this last group would have completed all signs.

WHERE Participants' drawings for WHERE were very consistent. All seven groups drew the 1-handshape symbol rotating about 20^o clockwise and counterclockwise. One group kept the arch symbol below the handshape symbol, which represented the movement trajectory in the (stationary) si5s notation. One group added two dots to upper-right corner of their frames, annotating their design "2 dots for repetition." Groups also differed in how many frames they drew.

UNDERSTAND Four groups drew the S-handshape symbol transitioning to the 1-handshape during an upward movement. Of these, one group added facial expressions, drawing flat eyebrows at the beginning and raised ones at the end, indicating that the sign is a question, as in "Understand?". Another of the four groups drew the index finger gradually extending, rather than jumping from one handshape symbol to the next. Two groups made their own notation with arrows pointing upwards.

MAYBE Five groups drew alternating vertical translations of the hand shape symbols, to indicate that the hands move up and down during the sign. One group changed the size of the hand shape symbols to indicate up/down movement, alternately drawing one handshape smaller than the other.

MOTIVATION Six groups again drew alternating vertical translations of the hand shape symbols, to represent the hands rubbing back and forth. Their notation differed from that for MAYBE in the direction of the thumb marks on the handshape notation (corresponding to a horizontal reflection of the handshape symbol). One of these groups rotated the handshape symbols to lie on their side, mimicking the hands with pinkies down during the live sign. Three groups added a line next to the pinkie of the handshape symbol to indicate that the pinkies point down, as in the stationary si5s notation. One group added two dots at the upper-right corner to indicate repetition.

Qualitative Feedback

Participants generally enjoyed the idea of creating an animated character system, several calling it "fun" and "creative." A few participants questioned the purpose of creating a new writing system, as they were used to using English or ASL video. Several wanted to continue the conversation beyond the end of the workshop, asking questions, discussing the design problems and opportunities,

and potential use cases.

During the written portion of the workshop, participants noted difficulties in designing animations. Several noted that it was difficult to represent certain aspects of ASL in the system, for example one participant noting "I think that the illustration of the hands' orientation in some cases could use a new way of expression; one that more clearly communicates the position of the hands." Participants also noted that it is difficult to decide on a single representation for a movement, raising questions such as whether the animation should gradually transition between handshapes or jump from one to the next.

Many students were interested in discussing potential use cases for an animated character system. One participant noted that he would like to use the system to help with finding ASL content. For example, he would like to enter a sentence in the animated writing system into a search engine, which would return matching portions of ASL videos. Other participants noted educational benefits of an animated writing system that demonstrates sign movements for people learning ASL, instead of complicated drawings of movements or stationary notation that does not convey movement. Other noted use cases included texting friends and taking notes.

5.4 Discussion and Future Work

The animation capabilities of computer graphics offer an exciting opportunity to create iconic character systems for sign languages. Existing writing systems have been limited to what is easily handwritten – stationary lines, curves, and dots – which fails to intuitively capture sign movements and consequently can be difficult to learn. An animated writing system represents sign movements directly, precluding the need to memorize complex movement notation. Animation also inherently depicts sign speed, a semantically meaningful aspect of ASL with gradations not easily captured symbolically. An intuitive character system could greatly benefit the Deaf community, whose primary language does not have a standard written form, and who have low literacy rates. Animated character systems could also be useful in other domains involving gesture, e.g., to document surgery procedures or the performing arts.

While we present the first prototype of an animated sign language character system, the work

presented in this chapter has several limitations. Our pilot study tested only a small set of signs. Though the set was chosen to span different types of movements, it was far from complete, leaving room for future studies with more comprehensive sets of signs. Our pilot study also evaluated recognizability of characters in isolation, whereas in daily life we typically encounter longer passages of text that provide context[8]. Still, the animated notation was more identifiable for individual signs, which is promising for learnability.

We also only explored a single framework for creating an animated character system – taking si5s, an existing stationary writing system, and replacing its movement characters with animation. Starting with an existing character system has the benefit of providing a hand-writable version. It is possible that animating a different character system, or building an animated system from scratch with animation in mind would produce a more favorable result. Creating an animated system from scratch is perhaps the most compelling, especially if we are not interested in preserving handwriteability, which becomes less crucial as computers become increasingly pervasive.

This work highlights an interesting trade-off between handwrite-ability and iconicity in sign language character systems. For example, it is possible to create an animated character system that can still be largely written, or has a hand-writable version (e.g., si5s and animated si5s). To provide the benefit of handwriting, such systems must be limited to monochromatic lines, curves, and dots, which limit the extent to which the text can resemble the live signing body. Alternatively, we can sacrifice the ability to write by hand in favor of creating truly iconic text involving multiple colors, complex shapes, and various animations. Many people currently write by hand, but it is not clear how valuable handwriting will be in the future, as we move away from paper resources towards electronic ones.[9]

Like all writing systems, sign language character systems abstract away some of the live language. For spoken languages like English, the community has agreed on which aspects of the language text must capture, and which may be left out. For a sign language character system to

[8]Unlike the pilot study, our design probe workshop *did* present a complete story

[9]Moving away from paper towards animated characters on screens raises many other issues, for example how to bridge the digital divide between people who have access to computerized devices and those who do not, and providing resilience to technical failures.

be widely accepted, the community must reach a similar consensus about which parts of the live language must be captured in text. Ambiguity in writing systems is also generally accepted. For example, heteronyms are words that are spelled the same but pronounced differently, e.g., “lead” (the metal) and “lead” (to guide). They are a subcategory of homographs, words with identical spelling but different meanings. Analogously, two signs might share notation but differ in meaning or execution. For a sign language character system to succeed, the community must reach agreement on acceptable ambiguities.

Because we are the first to propose creating an animated sign language character system, this work raises many new research questions. Our main research question was: Is there value in creating an animated ASL character system? This work suggests that the answer is “yes,” introducing vast future work including: exploring the design space of animated character systems further, examining the potential impact on literacy and childhood language development, developing teaching methods for animated character systems, exploring use cases for sign language classes, studying the effects of reading long passages of animated text, developing complementary input methods (e.g., an ASL keyboard), integrating animated scripts into existing text-based resources, providing online support for a community of users and contributers to the character system, and analyzing corpuses of animated text that users generate to spur developments in sign language translation and linguistics.

Design Dimension	The Number of Participants who Preferred Each Design					
1. X-Axis Movement	horz transl 	diag transl, sizing				other
	11	3				1
2. Y-Axis Movement	vert transl 	sizing 				other
	14	1				0
3. Z-Axis Movement	sizing	diag transl 	diag transl, sizing	vert transl		other
	12	2	1	0		0
4. Fingers up/down	180^o rot 	fading				other
	14	1				0
5. Toward/ away signer	horz refl 	horz refl, fading	vert refl			other
	10	3	1			1
6. Toward/ away center	180^o rot, stretch 	180^o rot	180^o rot, stretch, shadow			other
	8	5	1			1
7. Palm up/down	horz refl, stretch 	vert refl	horz refl, stretch, shadow			other
	9	3	0			3
8. Start of movement	no signal 	fade	pause	black dot marker		other
	5	4	3	2		1
9. Repeated movement	arrow, reps written 	dots above	inter-set fade	inter-set black dot	inter-set pause	other
	5	5	3	1	1	0
10. Display mode	single animated character displayed (left/right keys to progress)	full text displayed, animation on demand (on mouse hover)	full animated text displayed	full text displayed, sliding window animated (left/right keys to progress)		other
	8	5	2	0		0

Table 5.2: Participants' preferences on designs for each design dimension. (Viewed in Adobe Reader, table contains animations.) Key: diag: diagonal, horz: horizontal, vert: vertical, refl: reflection, rot: rotation, transl: translation, reps=repetitions.

Chapter 6

ASL-SEARCH AND ASL-FLASH

How can we build an ASL-to-English dictionary that is robust to query variability?

As explored in the previous chapter, text-based resources present accessibility challenges for ASL users. ASL does not have a standard written form that can be integrated into text-based platforms, often rendering these platforms unusable in ASL.[1] One fundamental informational resource that is traditionally text-based is the dictionary. Because dictionaries and search engines have predominately been designed for written languages, they provide poor support for looking up the meaning of unfamiliar signs. For instance, online web search resources[2] typically expect text input. Forming a search query for a sign is not intuitive, requiring a text description of a complex gesture or a guess at the English meaning.

Existing dictionaries designed specifically for ASL are not typically robust to query variability. For example, systems where users demonstrate signs to look them up require accurate execution of unknown signs, and ultimately do not work well, since vision and natural-language processing techniques are currently unable to accurately recognize or translate ASL into English. Instead of demonstrating the sign, feature-based ASL-to-English dictionaries allow people to use *features* of a sign (e.g. hand shape or location) to find the English meaning. Unfortunately, people must use the features expected by existing dictionaries, or they may not find the translation. The limitations of each approach demonstrate a need for a more intuitive and flexible dictionary system.

In this chapter, we propose ASL-Search and ASL-Flash, depicted in Figure 6.1. ASL-Search (Figure 6.1a) is a feature-based ASL-to-English dictionary powered entirely by its own users. Users form queries for unfamiliar signs by selecting sets of features (hand shapes, orientations, locations,

[1] The same problem exists for other sign languages.

[2] e.g. Google, Bing, YouTube, and online ASL video resources like SigningSavvy

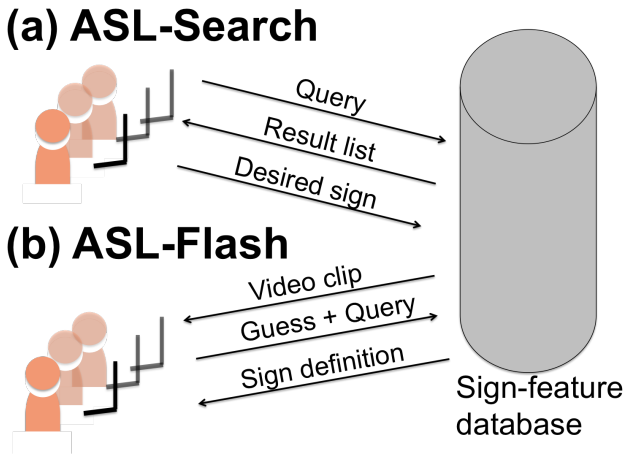


Figure 6.1: (a) ASL-Search: A feature-based dictionary that stores its users' queries in a database, and learns from that data to improve results. (b) ASL-Flash: An online learning tool that also contributes queries to the ASL-Search database. Users of both ASL-Search and ASL-Flash use the same query input interface, and contribute to the same database.

or movements) based on their observations. The dictionary is backed by a search engine that handles query variabilities by using Latent Semantic Analysis (LSA) on a database of user-provided feature evaluations. ASL-Flash (Figure 6.1b) is a learning tool for ASL students that provides an additional source of user queries for ASL-Search. ASL-Flash shows visitors flashcards of ASL signs. Before providing the user with the definition, ASL-Flash requests that the user acts as if he/she is searching for the sign using the ASL-Search interface. To validate our design, we deployed ASL-Flash with a 100-sign corpus, and simulated ASL-Search performance with the collected data.

The core contributions of this chapter are: 1) a survey of the methods that ASL students currently use to look up signs, and the difficulties they encounter, 2) design of ASL-Search, a user-powered ASL-to-English dictionary built by its users' queries, 3) ASL-Flash, a learning tool for ASL students that serves as a source for user query data, and 4) ASL-Search proof of concept with data gathered from ASL-Flash on a corpus of 100 signs.

6.1 Survey of Search Methods

To better understand problems encountered in looking up signs, we conducted a survey with current ASL students on the methods they use to search for unfamiliar signs. The survey used multiple choice questions to ask about the frequency of use for the resources in Figure 6.2, and free-form responses to gather more information on the students' lookup processes. We recruited 28 participants, 3 male and 25 female. The average (mean) age was 21, with a range of 18-41. All participants were either learning or already knew ASL, with a mean of 1.46 years of ASL experience. Two participants had experience with one additional sign language. The mean number of spoken languages including English was 1.89 per participant.

Despite the existence of online ASL-to-English dictionary resources, students predominantly did *not* use them. Each participant provided frequency of use for: 1) class textbook, 2) text ASL-to-English dictionaries, 3) other text resources, 4) online ASL-to-English dictionaries, and 5) other online resources. Examples of online resources that are not ASL-to-English dictionaries include YouTube videos, online English-to-ASL dictionaries, and search engines like Google or Bing. We found that the class textbook (92.86%) and non-dictionary online resources (75.00%) were the only resources used by the majority of students (see Figure 6.2). However, only 42.86% of the surveyed students used online ASL-to-English dictionaries. This disparity suggests that the internet is an appealing place for students to search for signs, but ASL-to-English dictionary performance or discoverability is not yet competitive with other online resources. Because most students use the internet to search for unfamiliar signs, it is possible that improved online dictionaries will provide benefit to those learning ASL.

Another valuable resource for inquiring about unfamiliar signs is to ask another person, but only 17 of the 28 participants used this method regularly (60.71%). One reason that fewer students ask others[3] is the lack of availability to do so. For example, it might be inconvenient or embarrassing to stop a conversation and ask, particularly in a social setting. In addition, there are other scenarios when it is never possible, like when viewing a video alone. The other 11 students (39.29%) who

[3] in comparison to using a textbook or online resource that is not a dictionary

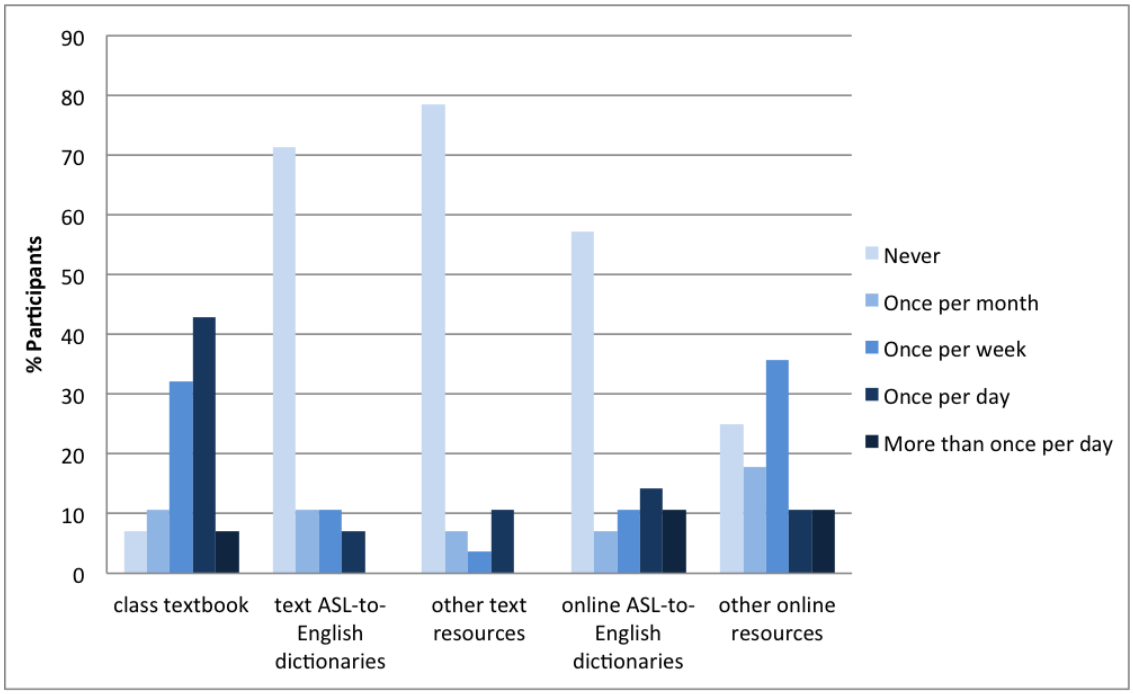


Figure 6.2: Use of existing sign search methods by ASL students.

did not ask others relied on text[4], online resources[5], and guessing the English representation.

Text and online resources as they exist today are not a viable solution to the problem of looking up signs for many students. In fact, 8 (28.57%) students said if they saw an unfamiliar sign, and had no one available to ask, they had no idea what actions to take. One possible reason is the difficulty of expressing a sign in text. In the words of one participant, *"describing the hand motions for a particular sign in words isn't always the easiest thing to do."* Another possible reason is that students must guess the meaning of the unknown sign using context in order to use English-to-ASL dictionaries, many text resources, or online videos. Other issues emerge when using ASL-to-English dictionaries; in particular, one participant alluded to inconsistent performance, explaining, *"I compare different dictionaries to make sure that I understand how the sign is formed or possible*

[4]class textbook, and other ASL books

[5]Google, Bing, YouTube, and AslPro [9]

variations on the sign." These responses strongly suggest that there is a need and opportunity to improve upon text and online resources when an ASL signer is not available.

In the face of these difficulties while searching for signs in online and text resources, 13 participants (46.43%) forego them altogether. If nobody is around to ask, 2 students (7.14%) ask somebody later and repeat the sign from memory. However, this task may be difficult because people may forget the need to ask, let alone the sign. 2 other students (7.14%) admitted that they were content with ignorance. *"I'd probably be content with just not knowing what the sign is,"* admitted one, while the other elaborated, *"It's such a visual language that trying to 'look' up how to sign a word can be more confusing than not having known it ever existed."* Several participants described a lineup of resources that they use. One participant explained, *"I first check my class textbook in the section where they go over vocabulary for the chapter. If it's not there I ask my twin brother who took the ASL series two years ago to see if he knows. If none of those things work I wait till it comes up in class to try and gain some more context for the sign."* The students' use of lists of resources highlights the difficultly and unreliability of searching for signs.

6.1.1 Implications from survey

Our survey demonstrated that current ASL learners not only struggle to search for the meanings of ASL signs, but stop their efforts altogether due to a lack of resources or motivation. Text and online resources are not conducive to inputting signs, and current ASL-to-English dictionaries are not complete or strong in performance. Therefore, our survey suggests that a usable, reliable, online ASL-to-English dictionary may remove an educational barrier: difficulty of searching for signs independently.

6.2 ASL-Search

We propose ASL-Search, an ASL-to-English dictionary that is entirely powered by its users. The dictionary allows users to search for a sign by selecting a set of features that describe the sign. The system stores queries from previous users in a matrix of feature frequencies for each sign.

As each user enters a query into ASL-Search, the system learns and improves the strength of its results using Latent Semantic Analysis (LSA). While the system's search interface allows an ASL sign to be described with many features spanning hand shape, orientation, location, and movement, LSA reduces the number of features needed (or dimensionality) when comparing a query to our database. By reducing the dimensionality of the feature space, LSA reduces noise in data while leveraging trends in the previously entered queries. For a feature-based sign language dictionary, there are several sources of variability in queries: 1) difficulty to create a comprehensive, unambiguous, and intuitive set of features that fully describe a signed language, 2) natural variability between different signers, 3) differences in perceptions between viewers, and 4) distorted memory of a sign. Instead of suffering from these sources of confusion, ASL-Search uses LSA to identify patterns in query variability and improve search results.

6.2.1 Search Interface

In order to design our search interface, we used the five feature types identified by Stokoe [140]. To be comprehensive, we used the hand shapes from the American Sign Language Handshape Dictionary (ASLHD) [150], while the rest of the features come directly from Stokoe notation. The features we chose are well grounded in linguistics, but further refinement of the features is an area for research outside the scope of this paper. The ASL-Search backend also compensates for imperfect design of the feature set by performing dimension reduction on the feature space. Below, we present each feature type, and the means to input features in a search query.

1. *Hand shape* (40 total): **What is the configuration of the hand and fingers?** Hand shapes are selected by clicking on pictures of the hand shapes. The pictures are of a fluent signer and organized in morphologic order, as in ASLHD.

2. *Location* (10 total): **Where relative to the body is the hand located?** The locations are selected by clicking on discrete regions of a picture of a person's torso and head.

3. *Orientation* (10 total): **Which direction is the palm facing? Are the fingers pointing up**

or down? The orientations are presented by a series of pictures of a hand facing in the appropriate directions.

4. *Movement* (22 total): **What is the change in position over the sign duration?** The movements are presented by ball-and-arrow diagrams of the hand movement. We also provide a video of a person making the movement on demand.

5. *Relative position* (7 Total): **If there are two hands, where are they located with respect to one another?** The relative positions are presented by images showing the relation between the two hands.

Our user interface allows for the selection of features to describe and search for a sign (as in Figure 6.3). We allow users to omit feature types, or to select multiple features within a single type. By giving the user freedom, the ASL-Search interface allows the user to describe the sign as he or she remembers. Next, we discuss how our backend uses LSA to compensate for the variability in search queries.

6.2.2 Backend

Latent Semantic Analysis (LSA) is traditionally used in Natural Language Processing to analyze the similarity between documents and words based on word counts [40]. In this domain, the data is represented as a matrix X. Each row represents a document, and each column represents a word. Item (i, j) thus contains the frequency of word j in document i. Instead of documents and words, our ASL dictionary has signs and features, as demonstrated in Figure 6.4. Each row of X represents a sign, and each column represents a feature. A query is represented as a vector of 0's and 1's, where a 1 indicates that a particular feature was entered as part of that query.

LSA utilizes Singular Value Decomposition (SVD) to factorize $X = U\Sigma V^T$, where U and V are orthogonal matrices, and Σ is a diagonal matrix. More specifically, U and V are matrices whose rows comprise the eigenvectors of XX^T and X^TX, respectively, while Σ contains the eigenvalues

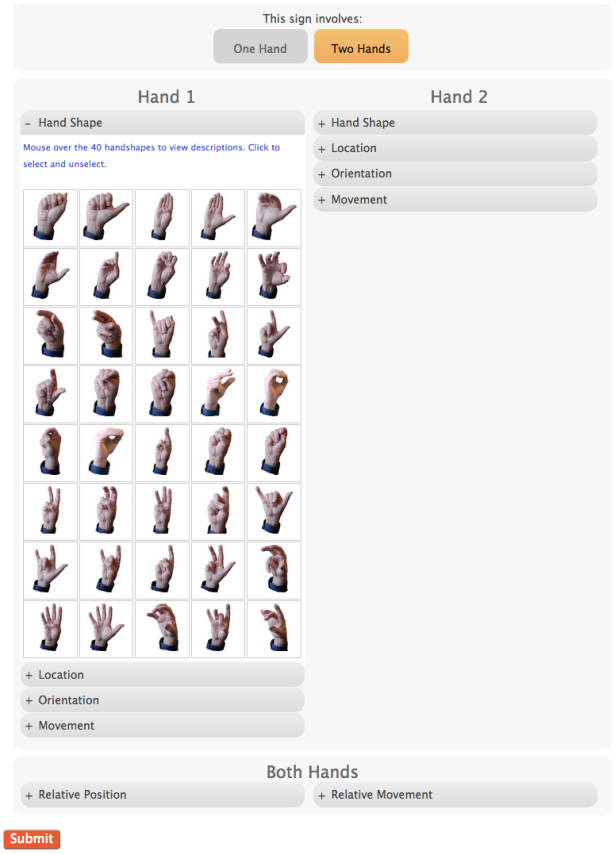


Figure 6.3: Screen shot of feature input interface as a user enters the hand shape for a two-handed sign. This interface is part of the ASL-Search design, and was deployed in ASL-Flash.

of XX^T (and also of X^TX). We can then select the k largest eigenvalues and corresponding eigenvectors to yield a rank-k approximation for X, $X_k = U_k\Sigma_k V_k^T$. This is an optimal approximation for X,[6] and can be used to map each feature (word) or sign (document) to an item in k-dimensional space.

To compute the similarity between an incoming query q and each sign in our database, we first project both q and X onto our lower-dimensional space as shown in Figure 6.5, yielding $q' = \Sigma_k^{-1}V_k^T q$ and $X' = \Sigma_k^{-1}V_k^T X$. We have now represented the query as well as each sign in our database

[6] optimal by the Frobenius norm

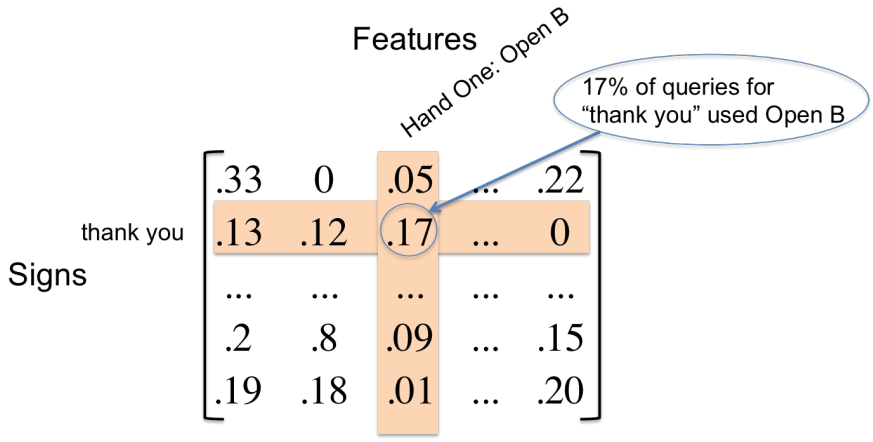


Figure 6.4: Matrix X of feature frequencies from user queries.

as a vector in k-dimensional space. We want to identify the signs in our database that are closest to the query vector, so we take the angle between the incoming query and each sign vector in the space of reduced dimensionality. More specifically, we return the signs in our database sorted by cosine similarity with the incoming query, computed as $\frac{x' \cdot q'}{||x'||||q'||}$, where x' is a projected sign from our database and q' is the projected query.

We tailored the classic LSA algorithm for ASL-Search to account for ambiguity between hands. When entering a two-handed sign, each user determines which hand to input as “Hand 1” and which as “Hand 2” (as in Figure 6.3). Because people can make two choices, we replicate all entered data by switching “Hand 1” and “Hand 2.” We store the flipped queries for each sign in a new row of the data matrix. A sign is a match for an incoming query when either the original or flipped row is a match. The intuition is that the flipped row more accurately represents the sign for users whose mental model of the two hands is opposite that of most users. By adding the flipped row, we can more accurately match queries for users who make the less popular hand choice.

6.3 ASL-Flash

We also present ASL-Flash, a learning tool that teaches students ASL and contributes queries to the ASL-Search database. ASL-Flash presents a series of ASL “flashcards” to the user. The sign

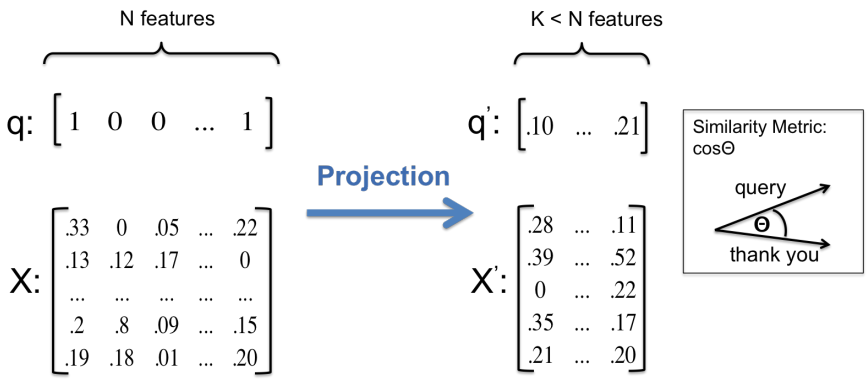


Figure 6.5: Dimension reduction from the original feature space to a lower-dimensional feature space.

clips are taken with permission from SigningSavvy [7]. Each clip shows a single sign. Once the user views the clip, ASL-Flash presents a multiple choice question on the English meaning. Next, the person is instructed to use ASL-Search's interface as if they were searching for the sign. Finally, ASL-Flash provides the English meaning of the sign accompanied by the original sign video.

Query data collected by ASL-Flash has several uses: 1) it will provide seed data for ASL-Search when the dictionary is deployed, 2) it will provide ASL-Search with query data for new signs added to the dictionary and existing signs that are looked up infrequently, and 3) it can be used to demonstrate the viability of the ASL-Search design, as we do in this paper.

6.3.1 ASL-Flash Deployment

We used ASL-Flash to gather query data to verify ASL-Search. Because ASL-Search is of use to people learning ASL, we used signs from a textbook used in first-year ASL curricula [175]. The textbook contains 1101 signs, and we selected a random sample of 100 signs to use in our deployment. Each ASL-Flash user viewed 10 of these 100 signs, randomly selected[8] and ordered. Users were given the option to quit early, so some respondents completed fewer than 10. Overall,

[7] http://www.signingsavvy.com

[8] from the least-viewed signs

we collected 670 viable queries from 94 users: 52 (55.32%) female, 41 (43.62%) male, and 1 (1.06%) other. The mean age was 28.4 years, with a range of 14 to 67 years. Users had a wide range of ASL experience (9 with 0 years; 14 with 0-0.5 years; 18 with 0.5-1 years; 14 with 1-2 years; 6 with 2-3 years; 32 with over 3 years; 1 chose not to respond). The mean number of signed languages known was 1.7 with a standard deviation of 3.7. The mean number of spoken languages was 1.9 with a standard deviation of 1.4.

ASL-Flash is a sustainable source of query data for ASL-Search. The sustainability of ASL-Flash is important because we will use ASL-Flash to seed ASL-Search upon release, and to gather data for new or rare signs. 79.72% of users said they would use the system again (76.92% of those who answered all flash cards correctly, and 81.25% with at least one wrong answer). Regardless of the user's accuracy in identifying signs, the tool provides value in learning or reviewing signs.

A strong motivation for people to continue using ASL-Flash (and providing more data for ASL-Search) are the learning benefits. Out of users who incorrectly guessed at least one sign, 87.50% reported learning something new from ASL-Flash. In addition, those same users showed more activity viewing the sign videos that accompany the sign definitions on the answer pages. The count of video views was not normally distributed for those who guessed all signs correctly (W = 0.4862, p-value < 0.001) or their counterparts (W = 0.7626, p-value < 0.001). Users who made mistakes viewed videos significantly more times than those who guessed all of the signs correctly (W = 158802, p-value < 0.001, Mann-Whitney). This suggests the dual benefit of ASL-Flash users learning ASL, while providing valuable data for ASL-Search.

During deployment, we asked users about their experience with the feature entry interface, which is identical to the interface proposed for ASL-Search. We found that our feature-based interface felt natural to most users. When asked the yes/no question: *"Did this survey give you a natural way to describe signs to look them up?"*, 75.68% said that the input mechanism was natural to use. Out of users who correctly guessed every sign, 73.08% found the interface intuitive; out of users who incorrectly guessed at least one sign, 77.08% agreed. This positive feedback suggests that the ASL-Search search interface will be suitable for looking up signs.

6.3.2 Feature Data from Experts

In addition to gathering features with ASL-Flash, we asked two experienced signers to provide features. They evaluated the same 100 signs used by ASL-Flash. Unlike in the ASL-Flash condition, we allowed the experts to replay the videos and complete the task in as much time as was needed. Their feature inputs were used to form baseline comparison search methods for ASL-Search.

The experts did not completely agree on the features present in the 100 signs. Though they agreed on the number of hands used for each sign, they entered slightly different feature sets for each of the 100 signs. On average (mean), they disagreed on 7.21 out of 164 features (4.40%). The lack of agreement between experts confirms the ambiguity inherent to sign executions and viewer perceptions. This ambiguity supports our choice of LSA for ASL-Search's backend, which extracts meaningful feature dimensions through dimension reduction.

The disagreement between experts also highlights the necessity of collaborative work in building a sign language dictionary. Even if the dictionary backend is constructed by experts (as in previous work), it would be inappropriate for a single expert to evaluate all signs; rather, a group of experts would be required. The tasks required of them would be time-consuming and tedious, and difficult to scale. Instead of dealing with these problems surrounding the collection and synthesis of expert input, our system supports the collaboration required to build an ASL dictionary effortlessly. The dictionary is built as it is used, by the users themselves.

6.4 Proof of Concept for ASL-Search Algorithm

We used the data gathered by ASL-Flash to form the database backend of ASL-Search and simulate its performance looking up signs. We formed baseline search methods that mimic existing ASL-to-English dictionaries using our expert feature evaluations. Our comparisons find that ASL-Search outperforms these baselines.

6.4.1 Metric

Discounted Cumulative Gain (DCG) is a standard metric used to evaluate the performance of search engines [74]. Let rel_i be the relevance of the i-th result. The score for a list of p ordered results is computed as $DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i}-1}{log_2(i+1)}$. For our problem, we are searching for one particular sign, so the relevance of each sign in the result list is binary $rel_i \in \{0,1\}$. Consequently, the DCG reduces to $\frac{1}{log_2(i+1)}$, where i is the placement of the desired sign in the result list.

Normalized Discounted Cumulative Gain (NDCG) normalizes DCG across queries, since the range of DCG scores that are possible varies for different queries. To do this, it divides the DCG for a given query by the maximal possible DCG for that query if results were returned in the optimal order. This means that NDCG will take a value in $[0,1]$ for all queries. For our problem, because relevance is binary and $rel_i = 1$ for exactly one sign i in the returned list, DCG and NDCG are equivalent.

6.4.2 Overall Performance

To validate the design of ASL-Search, we simulated its use with the query data gathered from ASL-Flash, as demonstrated in Figure 6.6. In the Testing Phase, we use leave-one-out cross-validation. The single held-out test query represents an incoming user query for a sign. We generate the sorted list of results that ASL-Search would return, using the rest of the data as its database of previous queries, and evaluate the quality of those results using DCG. In the Training Phase, we simulate ASL-Search learning the dimensionality to be used for reduction in the LSA algorithm. We run 10-fold cross-validation, computing a list of results for each query in the held-out part of the database, and average the DCG scores for each dimensionality. The dimensionality $K_{optimal}$ with the highest average DCG is chosen and used to generate results for the incoming test query.

Because our online dictionary returns a sorted list of results, we can identify the position of the desired result in that list. Figure 6.7 provides a histogram of those placements for the Test Phase. The desired result was the first result 59.10% of the time, and in the top 10 84.93% of the time.

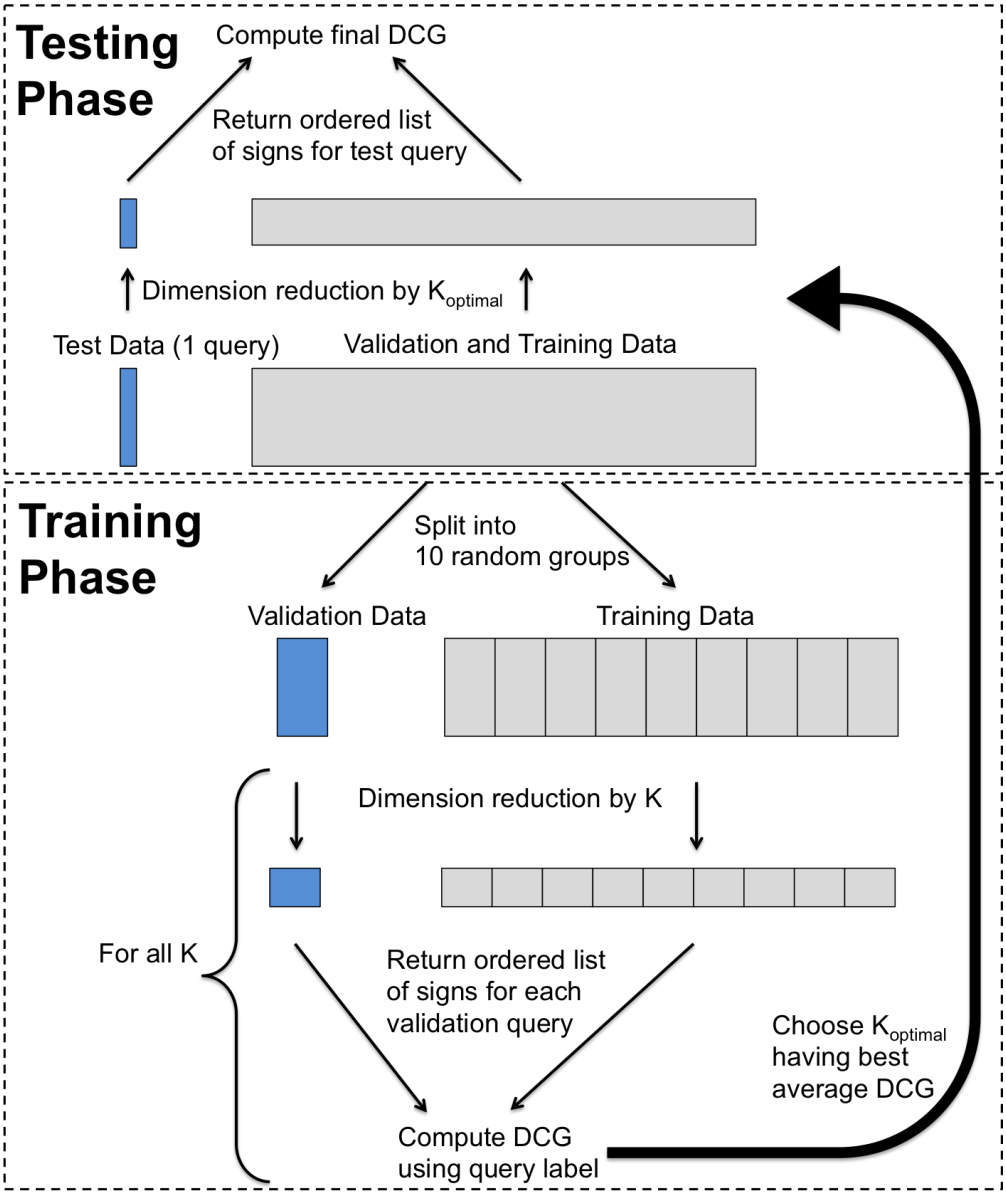


Figure 6.6: Simulated use and analysis of ASL-Search. The Training Phase replicates ASL-Search learning the dimensionality $K_{optimal}$ for reduction. The Testing Phase produces and evaluates the ordered list of signs that ASL-Search would return in response to incoming queries.

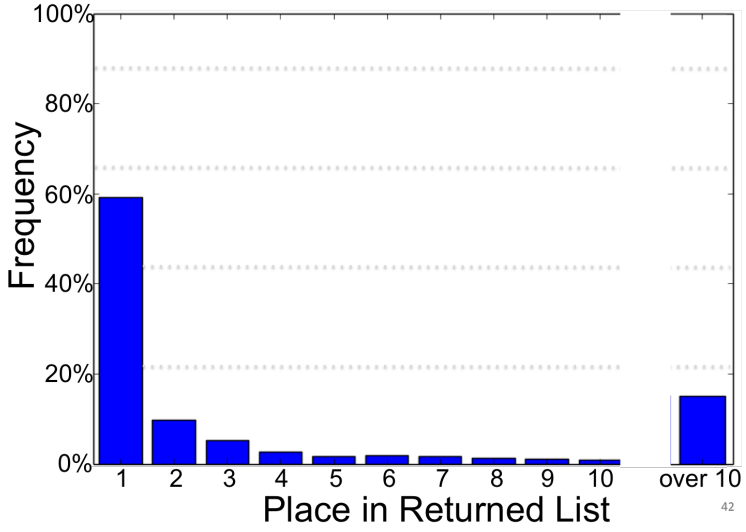


Figure 6.7: Place of desired result in the sorted result list. The "over 10" bucket summarizes the long tail of the distribution.

6.4.3 Tuning Dimensionality K

Choice of dimensionality to which the original feature space is reduced greatly impacts ASL-Search's performance, as demonstrated in Figure 6.8. Reducing to too few dimensions detracts from performance, as we lose meaningful information. Conversely, not reducing enough hurts performance, as we eliminate too little noise from the data. Because the choice of dimensionality impacts the quality of search results, ASL-Search uses cross-validation on the database of queries to learn the "best" choice of dimensionality for its users.

The dimensionality learned by the Training Phase was relatively stable throughout our simulation. Figure 6.9 shows the distribution of the chosen dimensionality. The mean optimal dimensionality found was 69.26. The feature dimensions produced by the dimension reduction are difficult to interpret intuitively. Each resulting dimension was a linear combination of all the original features.These results are characteristic of LSA, which is known to create dimensions that are difficult to interpret [85].

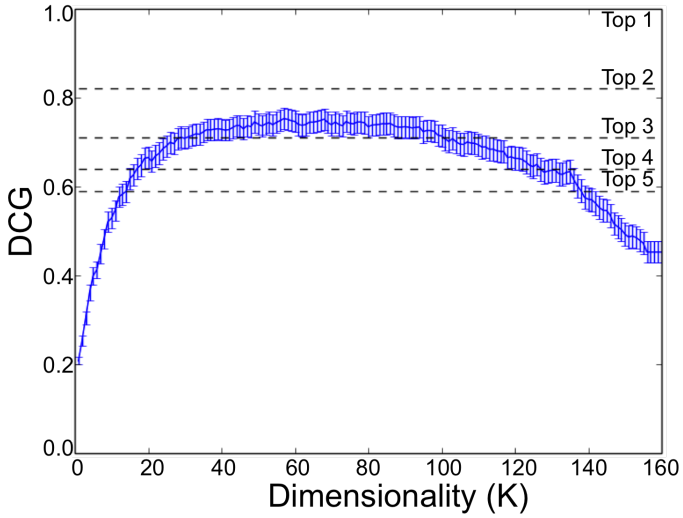


Figure 6.8: Effect of dimensionality K on performance, demonstrated by leave-one-out cross-validation on our entire dataset. The dotted lines show DCG when the desired sign is in the top 1, 2, 3, 4, or 5 results, with equal probability.

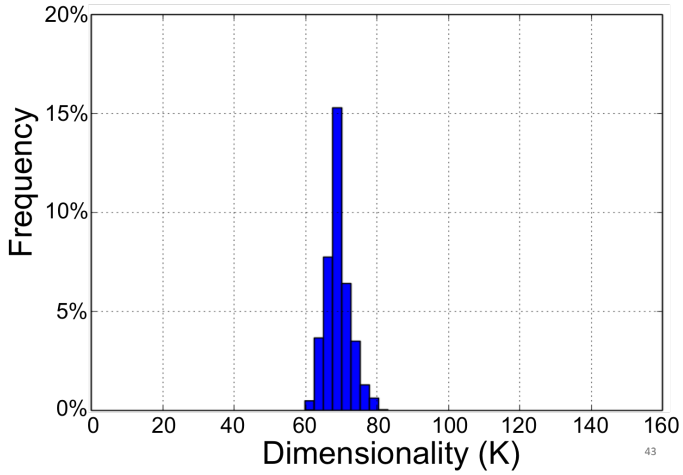


Figure 6.9: Dimensionality learned by ASL-Search ($K_{optimal}$ in Figure 6.6) from the database of past queries.

6.4.4 Baseline Methods

We compared ASL-Search to three baseline methods that return an ordered list of signs in response to a feature-based query. Because existing online ASL-to-English dictionaries do not seem to be fully functional and are not guaranteed to cover a vocabulary that matches the signs used in ASL-Flash, we generated our own baseline methods. These baselines are explained below:

- *Random*: Returns all signs in the dictionary in a completely random order.

- *Expert Or*: Compares the incoming query to the union (logical "or") of our expert feature vectors. It returns all signs for which the incoming features are a subset of the expert "or" vector, sorted by the number of matching features. We suspect that existing feature-based ASL-to-English electronic dictionaries use similar methods, since their results vary in length, and sometimes return no results at all. We chose the union of expert features, rather than their intersection (logical "and") because the union allows for more successful feature matches and better performance.

- *Expert Hamming*: Computes the Hamming distance between the incoming query and the union (logical "or") of our expert feature vectors for each sign in the dictionary. It returns all signs in the dictionary, ordered by increasing Hamming distance so that the closest signs are returned first. *Expert Hamming* serves as a more robust alternative to *Expert Or*.

6.4.5 Performance as the ASL-Search Database Grows

The simulated longitudinal performance of our dictionary demonstrates that ASL-Search significantly outperforms existing baselines. ASL-Search improves as more users contribute data, suggesting that ASL-Search will further outpace other methods in the future.

We simulated the performance of our dictionary over time using the query data gathered by ASL-Flash. We repeated the following two steps 20 times: 1) We held out a set of 100 test queries, one chosen randomly for each of our 100 signs. 2) We simulated the growth of the database

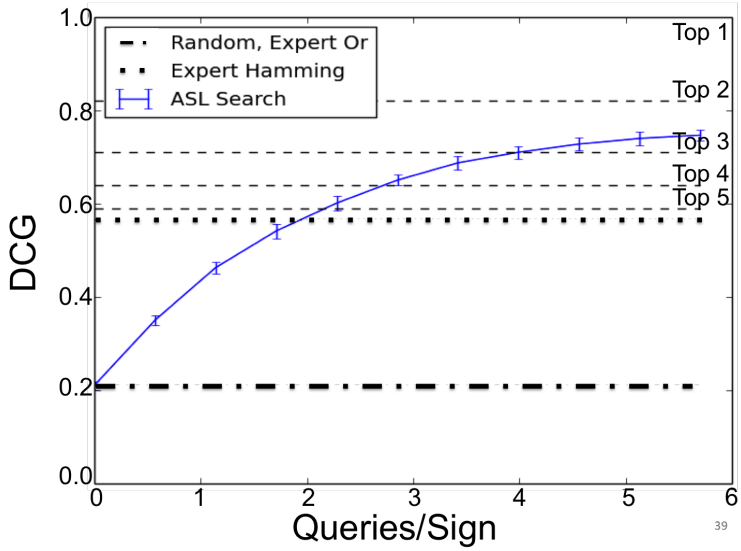


Figure 6.10: Performance of ASL-Search with use. Results were averaged over 20 simulations of dictionary growth.

by incrementally adding queries to the database, and evaluated performance on the test queries. Specifically, we divided the remaining query data into 10 equal sized and random groups, and added each group to the database to simulate database growth. Performance after each addition was evaluated by computing the DCG for the results generated for the 100 test queries. Figure 6.10 displays the average DCG from the 20 trials, with error bars of the standard error over the 20 trials.

Perhaps surprisingly, *Random* and *Expert Or* had almost the same average performance, with *Expert Or* outperforming *Random* by less than 0.001. The average performance of *Expert Or* is comparable to *Random* because it fails to return the desired sign for some queries. *Expert Or* only returns signs that the experts have determined to have all features selected in the query. Because there is variation in the features that viewers perceive, some queries contain features that the experts considered absent from the desired sign. In these cases, *Expert Or* does not return the desired sign (and in fact might not return any results at all), and receives a DCG score of 0. For these queries, *Random* outperforms *Expert Or* since it always returns the desired sign at some position in the result list. Conversely, for queries whose features are considered to be present in the desired sign

by the experts, *Expert Or* typically outperforms *Random*. While *Expert Or* is a weak baseline, it is representative of existing ASL-to-English dictionaries.

The *Expert Hamming* baseline better leverages the valuable signal in the expert data, and consequently outperforms *Expert Or*. Unlike *Expert Or*, *Expert Hamming* does not restrict the signs returned to those that possess all features present in the query. Instead, it uses Hamming distance to evaluate similarity between the incoming query and every expert evaluation of a sign in the dictionary. Using Hamming distance produces more nuanced comparisons, and allows *Expert Hamming* to always return the desired sign at some rank in the result list.

Overall, our system returned more accurate results than the *Random*, *Expert Or*, and *Expert Hamming* baseline methods. The crowd provides data that experts do not: the features commonly seen by real users who are unfamiliar with the signs they look up. ASL-Search's use of dimension reduction through LSA allows ASL-Search to leverage this signal. Furthermore, our system improves with additional data from users, which is not possible for *Random*, *Expert Or*, *Expert Hamming*, or variants of our baselines employed by existing online ASL-to-English dictionaries.

6.4.6 Performance by ASL Experience

Our system performs well for target users with varying levels of ASL experience, as seen in Figure 6.11. To generate these results, we set the dimensionality to 70, which was the mean dimensionality chosen in the Training Phase, and held out one query at a time for testing. Our performance is relatively stable for target users of varying levels of ASL experience, but was slightly worse for those with absolutely no ASL experience. One possible reason for this lower performance is that brand new signers may have trouble identifying the visual queues in a sign. They may have suffered from fatigue as well, with decreased quality as they progressed through the flash cards.

Our proposed system typically returned the most appropriate results when the English meaning of the sign was not known, as shown in Figure 6.12. For all levels of experience, except for users with 1-2 years of ASL, queries for unknown signs produced better results than those for known signs. Even for users with 1-2 years of ASL, the difference in performance is negligible. It is likely that LSA handles queries for unfamiliar signs better because these queries are characterized by

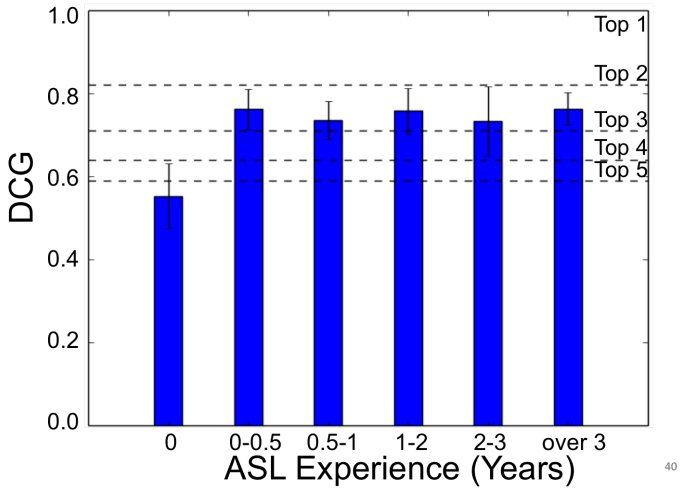


Figure 6.11: Performance for queries entered by users with varying ASL experience.

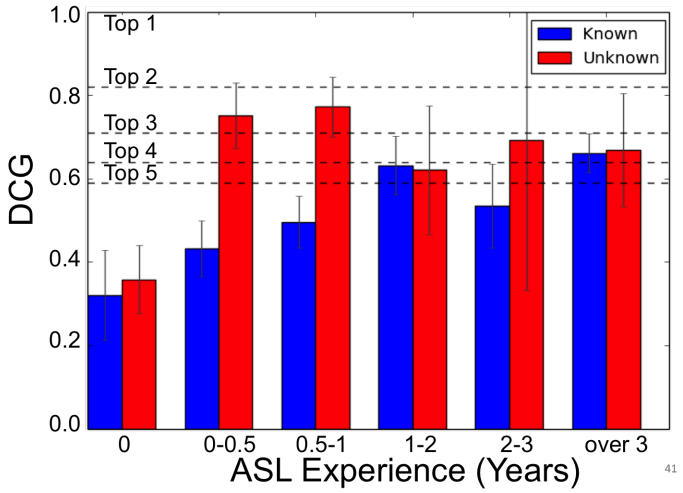


Figure 6.12: Performance for queries entered by target users with varying ASL experience, separated by whether or not the sign was known.

feature variabilities that LSA expects and compensates for. Our results suggest that ASL-Search should fulfill its purpose of helping users to look up ASL signs, and especially unfamiliar ones.

6.5 Discussion and Future Work

This chapter presents the design of two novel data-driven systems that support information access for ASL students in distinct but related ways: 1) ASL-Search, a feature-based electronic ASL-to-English dictionary that is both easy to use and accurate, and 2) ASL-Flash, a set of online video flashcards that help students learn ASL. The two systems work together to support the community of ASL learners and users. As ASL-Flash users reinforce and acquire ASL knowledge, they also contribute query data that improves the performance of the ASL-Search dictionary.

The survey we ran on existing lookup methods exposed difficulties that ASL students encounter when searching for a sign definition, and highlighted the need for a tool that allows users to easily and accurately look up ASL signs. Our proof of concept suggests that ASL-Search can accurately match queries to signs, and will improve with use, eventually outperforming existing baseline methods that do not improve with use. Our deployment of ASL-Flash with a small corpus of signs demonstrated that ASL-Flash is both an effective data collection tool for ASL-Search and a valuable learning tool in its own right.

A dictionary that learns from a crowd of self-motivated ASL users (through ASL-Search queries and ASL-Flash feature quiz responses) has several advantages. Its language model is robust to common mistakes, handles linguistic variations such as regional dialects, and evolves as the language changes. For example, if the hand shape used for a particular sign is commonly mistaken, varies geographically, or changes over time, ASL-Search will model those variations in its users' queries, and adapt its results accordingly. Furthermore, because its training data is a by-product of people looking up signs or studying ASL vocabulary, providing data is not a burden.

Our dictionary design can be applied outside of ASL, to other sign languages and new domains. Signs in any sign language are three-dimensional motions not easily described with written words. Their execution and perception are characterized by natural variability, making our LSA-backed dictionary design relevant. Releases of ASL-Search and ASL-Flash tailored to other sign languages

besides ASL could benefit a larger community of signers. Our system can also be adapted to support search for other types of items not easily described in written queries besides signs. For example, a version of ASL-Search could support bird watchers looking up bird species they see (and plant/animal taxonomies more generally). The bird can be described by a discrete set of features, like feather color and beak shape. LSA is well suited for this domain because of natural variability in people's feature selections.

Chapter 7

ASL-SEARCH AND ASL-FLASH SCALABILITY EXPLORATION

How well does our ASL-to-English dictionary design perform at scale? How can we change our design to better support scalability?

The previous chapter presented an ASL-to-English dictionary designed to be robust to query variability due to differences in sign execution and perception. While the proof of concept suggested that the dictionary design can handle variability in queries, the evaluation was run on a small corpus of 100 signs. A functional dictionary must scale to thousands of signs. Even introductory vocabulary taught in the first two years of ASL courses covers over 1,000 signs.[1] Retrieving accurate matches over a larger corpus of signs introduces more opportunities for dictionaries to make mistakes by returning undesirable signs similar to the desired sign. The goal is for our dictionary design to perform well at scale despite the increased difficulty.

This chapter explores how well our dictionary design performs at scale, both in terms of ASL-Flash's ability to collect data and ASL-Search's performance over a large corpus of signs. To do this, we deployed a scaled version of ASL-Flash with 1,161 basic signs, and analyzed the data quality collected as well as the simulated performance of ASL-Search based on that collected data. Key contributions of this chapter are: 1) a scaled ASL-Flash design that addresses challenges of corpus coverage and participant incentivization, 2) deployment of this new design to gather a scaled corpus of feature evaluations, and 3) an analysis of the viability of both ASL-Flash and ASL-Search at scale based on the collected data.

---

[1]The size of the ASL lexicon has not been established, and is difficult to estimate because signs can be modulated in various ways to provide nuanced meanings. As an approximate lower bound, taking all combinations of Stokoe features for a single hand, with one feature per feature type, yields 40 handshapes x 10 locations x 10 orientations x 15 movements $\approx$ 60,000 one-handed signs.

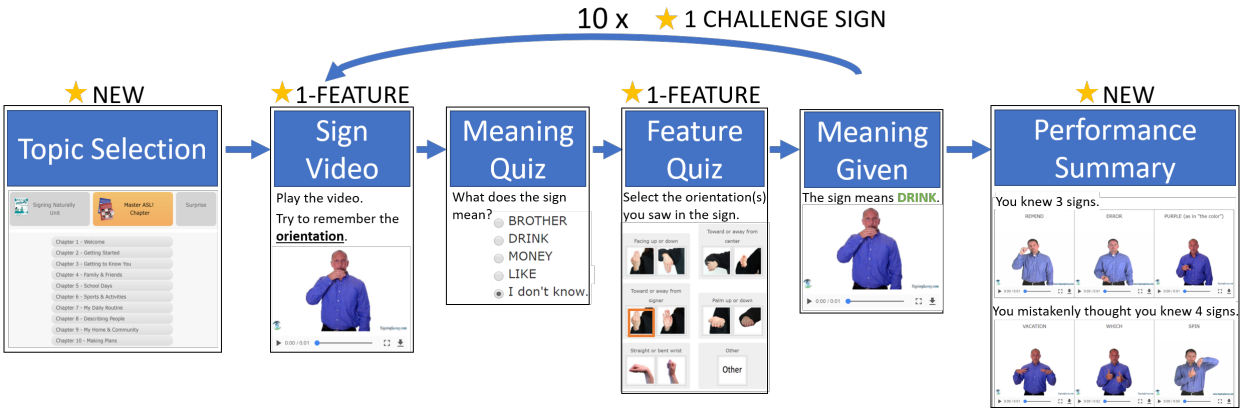


Figure 7.1: Scaled ASL-Flash design. Starred modules are new or modified to support data collection at scale.

7.1 Scaled ASL-Flash Design

In order for ASL-Flash to provide seed data for ASL-Search at scale, the site must 1) be monetarily sustainable, 2) provide a sufficient throughput of high-quality feature evaluations, and 3) provide coverage (feature evaluations) of the full sign corpus. This section describes the major design changes made to satisfy these conditions, with the new design summarized in Figure 7.1.

7.1.1 Participant Incentives and Compensation

Our new design makes it possible for users to study specific content taught in their ASL courses. The original ASL-Flash design incentivized participation through intrinsic educational benefits derived from tasks, but the proof-of-concept deployment used additional monetary compensation to ensure adequate data collection. To incentivize students to use ASL-Flash without payment, we re-designed the site to more closely align with the objectives of students in ASL courses.

We updated the signs that the site supports to include signs commonly taught in introductory ASL courses. Specifically, we include all signs from two popular ASL textbook series, Signing Naturally [96, 97] and Master ASL! [176]. These textbooks cover approximately the first two years of language courses. They span 1,161 distinct signs, 707 from Signing Naturally and 768

from Master ASL! (314 overlapping). The list of signs in the textbooks were compiled and matched with sign videos from SigningSavvy by a fluent ASL signer.

We re-designed the platform to give participants control over which signs they review (and provide data for). Specifically, we organize signs by the two popular ASL textbooks above, and allow users to choose a book and chapter to study (Figure 7.1 Topic Selection module). The signs chosen during their session on the site are then chosen from that book chapter. At the end of the session, we provide a summary of the user's performance to help the user assess their progress and memorize corrections (Figure 7.1 Performance Summary module). The summary provides the video and English meaning for each sign studied, organized into three sections: 1) signs they correctly identified, 2) signs they incorrectly identified, and 3) signs where they selected "I don't know" instead of identifying the meaning.

7.1.2 Task Design

Our modified feature evaluation task reduces cognitive load to encourage task completion. The original ASL-Flash flashcard task consisted of the participant viewing a sign video, a quiz on its English meaning, and a quiz on *all* its feature types (handshape, location, movement, and orientation). The feature evaluation task (choosing from a series of drop-down grids with 175 selectable images) was much more involved than the meaning quiz task (choosing from a set of 5 radio buttons). The high cognitive load to evaluate 175 possible features is likely to discourage feature evaluation completion.

To reduce cognitive load, we revised the flashcards to focus on a single feature type for each flashcard. The instructions for playing the sign video specify which feature type the participant should remember. The feature quiz then asks the participant to evaluate that feature type, providing a drop-down menu to select evaluations for that feature type only. All other feature evaluations are optional. We provide an "Input More Features" button at the bottom, which when clicked, triggers drop-down menus for other feature types to appear. The feature type for each flashcard is chosen randomly. We also add an "other" feature option to each feature type to provide more comprehensive answer options, increasing the total number of features from 165 in the original

design to 175.

7.1.3 Question Selection

Giving users control over which signs they study (and which tasks they complete) introduces the problem of ensuring coverage. In particular, it is likely that more ASL-Flash users will be interested in studying early textbook chapters, since some students drop out of courses or otherwise choose to stop studying the language. As a result, we might expect to accumulate more feature evaluations for signs that appear in these chapters.

To ensure that we generate feature evaluations for our entire corpus, we incorporate a "challenge" sign from an advanced chapter lacking data within each practice session. ASL-Flash feeds participants signs in batches of ten. One of these ten signs is chosen randomly from the set of signs with the least data. To prevent discouragement, the sign is clearly labeled as a challenge sign, and an encouraging tooltip. We also give users the option of reviewing a Surprise set of signs (Figure 7.1 Topic Selection module), which the system chooses to be those with the least data. This design still provides educational value by building vocabulary, while also helping to provide data on scarce signs.

7.2 Preliminary Scalability Study

To explore the efficacy of our scaled design, we deployed the scaled ASL-Flash design. Based on the data collected, we analyze 1) the quality of the large-scale feature evaluations generated and 2) the expected performance of the ASL-Search dictionary at scale.

7.2.1 Method

We deployed the scaled ASL-Flash design as a publicly available website with IRB approval. It ran for three years, collecting feature evaluations for the scaled corpus of 1161 signs described in the design. The site gave participants the option to create accounts and enter basic demographics.

For comparison, we collected expert feature evaluations of the scaled corpus. To do this, we

built a website that provided a list of all 1161 signs. When the expert clicked on a sign, they were taken to a feature input page for that sign, consisting of a video of the sign at the top that could be replayed, and the complete feature input interface for all feature types. The expert could return to a sign to update their evaluations at any time. A single fluent ASL user provided complete feature evaluations for the full corpus.

Simulation of the ASL-Search dictionary at scale was accomplished by training and testing the dictionary on separate sets of feature evaluations, as in the proof of concept in the previous chapter (see Figure 6.6). For all simulations, we tested the dictionary on feature evaluations collected in our proof-of-concept deployment of ASL-Flash. These feature evaluations best mimic ASL-Search queries since the feature evaluation task mimicked looking up a sign in the ASL-Search dictionary, prompting the user "The sign involves:" and providing the full feature input interface. They were also randomly selected and cover a range of sign complexity. We experimented with training the dictionary on different data sources, described below.

7.2.2 Datasets

In our ASL-Search simulations, we used several different datasets, outlined below:

- **Flash**$_{PC}$: the corpus of feature evaluations collected for 100 random signs through our proof-of-concept (PC) deployment of ASL-Flash, described in the previous chapter
- **Flash**$_{SCALED}$: the corpus of feature evaluations collected on 1,161 introductory-level signs through deployment of the scaled ASL-Flash design, described in this chapter
- **Expert**$_{SCALED}$: the corpus of feature evaluations over our 1,161 provided by our ASL expert, described in this chapter

We also experimented with datasets formed by combining the datasets outlined above:

- **Flash**$_{SCALED}$**+Flash**$_{PC}$: a corpus that is the aggregation of Flash$_{SCALED}$ and Flash$_{PC}$
- **Flash**$_{SCALED}$**+Expert**$_{SCALED}$: a corpus that is the aggregation of Flash$_{SCALED}$ and Expert$_{SCALED}$

7.2.3 Metrics

To evaluate the quality of data collected by the scaled ASL-Flash platform, we compute basic statistics on the data, including feature evaluation recall and distribution of data over signs.

To evaluate how well the ASL-Search dictionary design works when backed by the scaled ASL-Flash data, we use Discounted Cumulative Gain (DCG) [74]. DCG is a standard search engine metric, and was used in the initial small-scale ASL-Search proof of concept. It evaluates how well results (a sorted list of signs) match incoming queries (feature sets).

7.2.4 Participants

Participants were recruited through relevant email lists, social media and web posts. We had 765 unique site visitors provide feature data. 202 of these participants created profiles for themselves, providing us with demographic information. Their ages ranged 13-75, with a mean of 37. The gender breakdown was: 154 (76%) female, 42 (21%) male, 4 (2%) other, 2 (1%) unprovided. Their level of ASL experience was:

Years ASL	# Participants	% Participants
0	36	18%
0-.5	28	14%
.5-1	40	20%
1-2	35	17%
2-3	11	5%
>3	51	25%
(unprovided)	1	<1%

7.3 Preliminary Results: ASL-Flash at Scale

The quality of the data collected by the scaled ASL-Flash design, analyzed in this section, suggests that our design encouraged participants to provide feature evaluations, and can attain coverage of a complete scaled corpus.

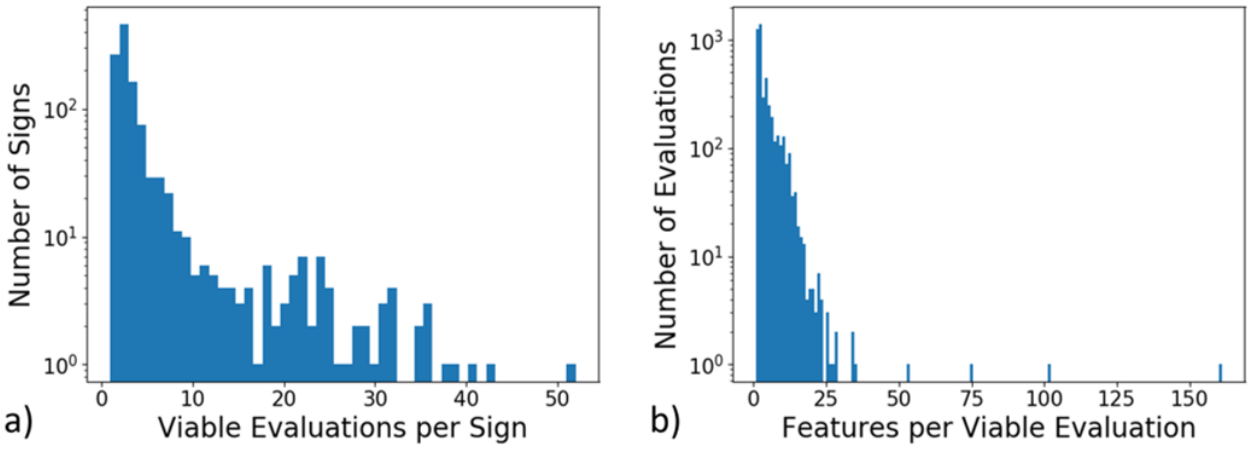


Figure 7.2: Distribution of feature evaluations collected on a log-scale in terms of a) viable feature evaluations per sign, and b) the number of feature selections per evaluation.

7.3.1 Data Coverage

The scaled ASL-Flash site collected 11,400 feature quiz responses, 4,633 of which were viable, meaning they contained features beyond whether the sign involved 1 or 2 hands (which was required). At least one viable feature quiz response was provided for each sign. The distribution of data is shown in Figure 7.2, both in terms of feature inputs per sign, and feature inputs per feature quiz. The distribution of data over the signs is heavily skewed left, with the vast majority of signs having under 5 feature evaluations, highlighting the need for strategically chosen "challenge" signs to even out the distribution.

7.3.2 Feature Evaluation Recall

To evaluate the effect of simplifying the feature quiz to focus on a single feature type, we deployed the site both with and without this design modification. Our results (Figure 7.3) show that our scaled design elicited evaluations 50% of the time (up from 32%), suggesting that simplifying the task increased how many people input features and encouraged feature input across feature types. These results provide evidence of the generalizability of findings that short task time and low

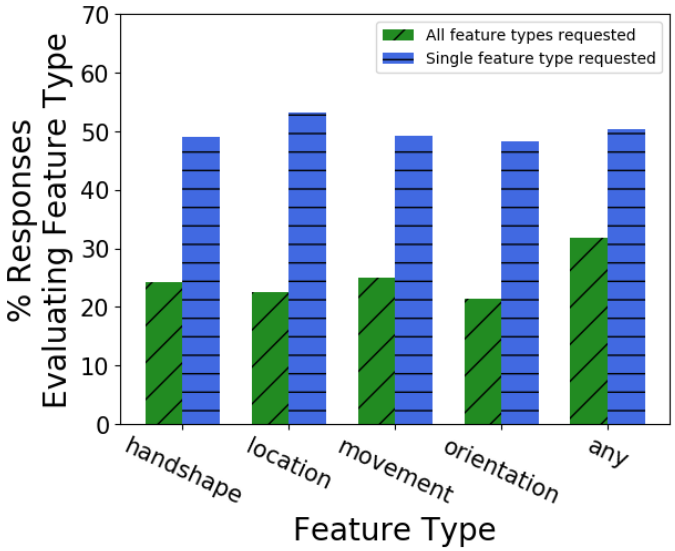


Figure 7.3: Percent of feature quiz responses that were viable (contained feature evaluations, beyond the required 1 vs. 2 hands). Recall increased when a single feature type was requested regardless of which feature type was requested. Overall, limiting the feature quiz to one feature type increased recall by 58% (the "any" Feature Type).

cognitive load are required to collect high-quality crowdsourced data (e.g., [28, 151]). While past work has focused on crowdsourcing platforms that provide monetary compensation, our results generalize beyond monetary payment to include platforms that incentivize participation by aligning with other motivations, in our case education.

7.4 Preliminary Results: ASL-Search at Scale

A larger corpus of signs introduces more opportunities for a dictionary to return undesirable matches to an incoming query. In this section, we explore ASL-Search's expected performance when scaled to introductory ASL vocabulary, by simulating its performance when trained on Flash$_{SCALED}$, the feature evaluations collected through the scaled ASL-Flash deployment. The simulations suggest that accurate performance over this larger corpus of 1,161 signs requires more data than has currently been collected by the scaled ASL-Flash.

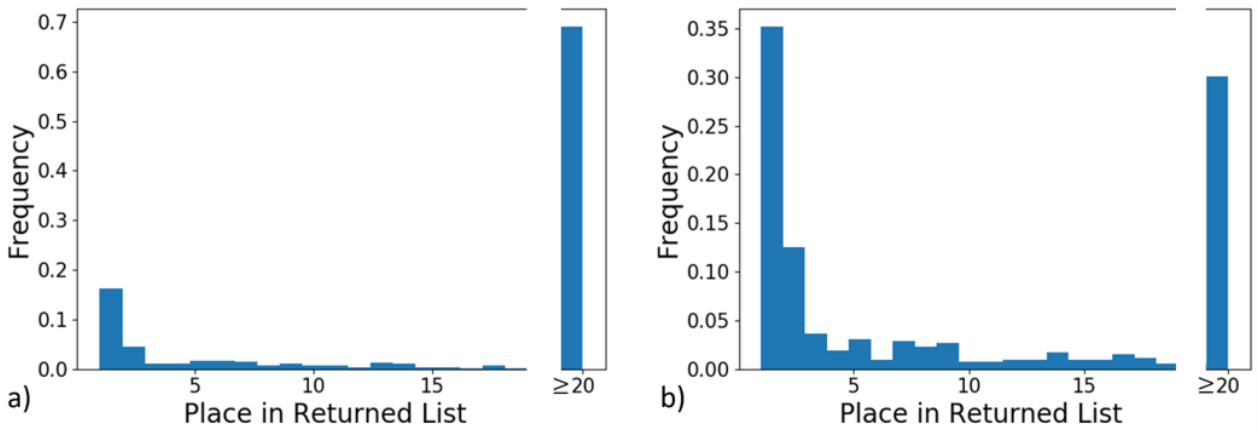


Figure 7.4: ASL-Search's expected performance at scale (searching over 1,161 signs) when trained on a) Flash$_{SCALED}$ alone, and b) Flash$_{SCALED}$+Flash$_{PC}$ (excluding the test query). Test queries are taken from Flash$_{PC}$. The histograms show the place of the desired sign in ASL-Search's sorted result list. The ≥ 20 bucket summarizes the distribution's long tail.

7.4.1 Overall Performance

The dictionary's simulated performance when searching over our 1,161-sign corpus is shown in Figure 7.4, in terms of the distribution of the desired sign's rank in the results list. Performance searching over 1,161 signs is significantly lower than in our 100-sign proof of concept, with almost 70% of desired results falling in the ≥ 20 bucket when trained on Flash$_{SCALED}$ alone (Figure 7.4a), compared to under 20% being in the 10^{th} place or later in the 100-sign proof of concept (Figure 6.7). The significantly improved performance to only about 30% of signs falling in the ≥ 20 bucket when Flash$_{PC}$ is added to the training set (Figure 7.4b vs. 7.4a) suggests that the dictionary would benefit from collecting more data before deployment. While lower performance with a larger corpus of signs is expected, it is not clear what level of accuracy would be acceptable to users. Exploring this question through user studies is an area for future work.

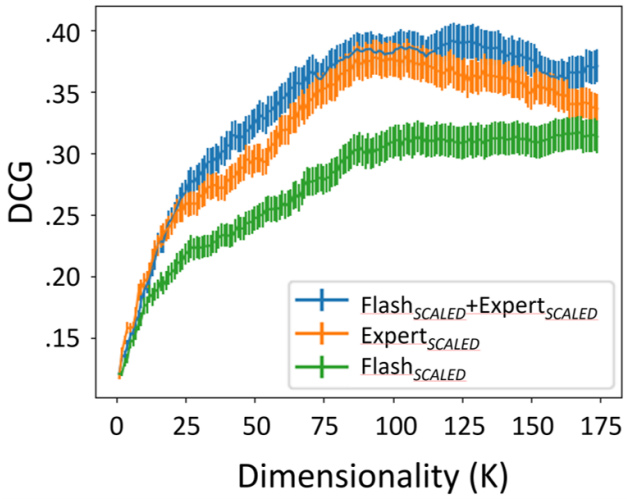


Figure 7.5: Simulated performance of ASL-Search over a corpus of 1,161 signs when trained on Flash$_{SCALED}$, Expert$_{SCALED}$, and Flash$_{SCALED}$+Expert$_{SCALED}$.

7.4.2 Comparing Training Sources

We experimented with training the dictionary on three sources: 1) Flash$_{SCALED}$, data collected by the scaled ASL-Flash deployment, 2) Expert$_{SCALED}$, our expert feature evaluations, and 3) Flash$_{SCALED}$+Expert$_{SCALED}$, data collected by the scaled ASL-Flash deployment combined with our expert feature evaluations. For the third source, the expert evaluations received equivalent weight to each of our ASL-Flash participants' evaluations. For each training source, we tuned dimensionality K used in LSA's singular value decomposition step.

The results of our simulation on different data sources and dimensionalities is presented in Figure 7.5. Training on Flash$_{SCALED}$+Expert$_{SCALED}$ performed the best overall, followed by Expert$_{SCALED}$, followed by Flash$_{SCALED}$. In our proof-of-concept with only 100 signs, our dictionary performance peaked at a DCG of 0.7 with optimized dimensionality K, while in this scaled analysis, the dictionary peaks at a significantly lower DCG of almost 0.4. It is also worth noting that performance based on Flash$_{SCALED}$ alone continues to climb as dimensionality K increases, again indicating that we have not yet gathered enough data when training only on this source.

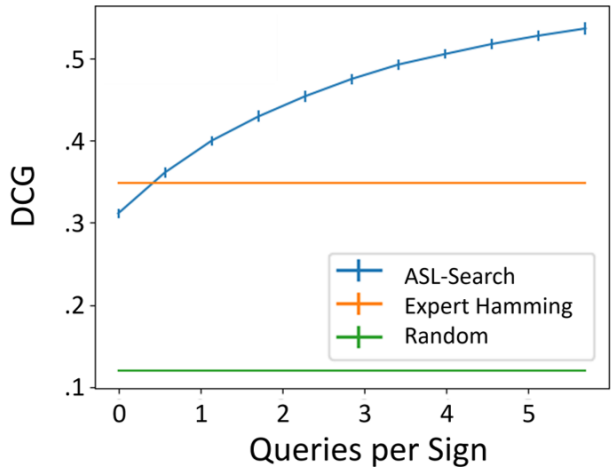


Figure 7.6: Expected improvement of the dictionary as it adds queries from dictionary users (simulated by feature evaluations from Flash$_{PC}$) to its initial LSA backend model (simulated with Flash$_{SCALED}$), compared to Expert Hamming and Random baselines. Test queries are taken from Flash$_{PC}$ (and do not overlap with training queries).

7.4.3 Expected Performance over Time

We also experimented with gradually adding complete queries (feature evaluations from Flash$_{PC}$) to the training data (feature evaluations from Flash$_{SCALED}$) to see how performance might improve once the dictionary is released. Figure 7.6 shows the results, compared to our best baseline from our proof of concept, Expert Hamming, which returns signs based on Hamming distance between the incoming query (feature evaluations) and the expert's feature evaluations of all signs in the corpus, as well Random, a completely random ordering of results. As in our smaller proof of concept, performance improves as queries accumulate. The improvement in performance, coupled with the current expectation that Expert Hamming would outperform our dictionary at deployment, reinforce the suggestion that our model would benefit from further data collection through ASL-Flash before deployment.

7.5 Discussion and Future Work

This chapter presents a scaled ASL-Flash design, which we deployed to collect data and simulate ASL-Search performance at scale. Our ASL-Flash deployment suggests that our scaled ASL-Flash design can successfully elicit feature evaluations for a large corpus of signs, and is an engaging educational resource for students. Our simulations of ASL-Search at scale, using feature evaluations collected through our scaled ASL-Flash deployment, suggest that our ASL-Search dictionary design outperforms existing baselines with sufficient data, which has not yet been collected.

We plan to continue improving the performance of the ASL-Search dictionary at scale until it outperforms existing baselines, at which point we will publicly release the dictionary. To do this, we will continue collecting data through our scaled ASL-Flash design, and rerun our simulations periodically to gauge how much more data is needed. We will also experiment with customizing our lookup method based on the new type of data collected by ASL-Flash. Specifically, the scaled ASL-Flash design elicits evaluations of just one feature type at a time. It is possible that simply aggregating all feature evaluations for each sign (as LSA does) is no longer appropriate. Instead, we could experiment with reducing the weight for feature types that were frequently elicited for a particular sign, or increasing the weight for feature types that yielded high recall (feature types that participants frequently voluntarily evaluated).

We also plan to investigate other algorithmic ways to improve search performance. Alternative topic modeling methods could be used, including probabilistic latent semantic analysis (PLSA) or latent Dirichlet allocation (LDA) to model the co-occurrence probability of queries and signs. Alternate feature-sign weights, such as term frequency-inverse document frequency (TF-IDF), and other dimension reduction techniques, like principal component analysis (PCA) or group sparsity methods, could also be used. With or without dimensionality reduction, classification methods like decision trees, support vector machines (SVMs), and multinomial logistic regression could also be used to determine the sign class for the incoming query, instead of LSA's cosine similarity. Unsupervised or semi-supervised methods could leverage unlabeled data, for example queries from ASL-Search where the user did not provide information about which sign they sought. Relevant

methods can also be combined in various ways, for example by applying a sequence of operations, or weighting results from methods run in parallel.

The planned release of the ASL-Search dictionary also introduces new research directions. For example, the dictionary must determine when the user finds their desired sign, so their query can be added to the dictionary's backend model of the appropriate sign. If the user does not explicitly mark the found sign, determining the sign can be difficult. Methods such as tracking which results are visited should be explored. The opportunity to incorporate negative feedback (indications that signs returned by the dictionary are *not* desired) also arises. For example, they query features could be subtracted from the dictionary's model of top-ranking irrelevant signs to correct for the mistakes. The question of how the dictionary should handle queries for signs not in the dictionary should also be addressed. Is it possible to accurately determine that the desired sign is not in the dictionary corpus, for example using a confidence score of returned results? If so, how should the user be notified? We look forward to answering these questions, and seeing the full system deployed for public use.

Chapter 8

CONCLUSION

This dissertation addressed challenges in information access for low-vision readers and sign language users, by leveraging modern computing capabilities to redesign informational resources. In this section, I review the major contributions, present a vision for continuing this line of research, and highlight key takeaways.

8.1 Summary of Contributions

In this dissertation, I presented five main systems that improve information access for low-vision readers and sign language users by leveraging modern computing capabilities. In designing, building, and evaluating these systems, I employed tools from computer graphics, crowdsourcing, topic modeling, and participatory design:

1. Smartfonts, alternate English scripts that use computer graphics to redesign letterforms, thereby improving legibility, with designs informed by crowdsourced perceptual data

2. Livefonts, a type of Smartfont that introduces the use of animation to differentiate characters in a script, further improving legibility

3. Animated Si5s, the first animated ASL character system prototype, designed to resemble live signing, making it easier to learn and use

4. ASL-Search, a scalable feature-based ASL-to-English dictionary that is robust to query variability, powered by crowdsourced feature evaluations

5. ASL-Flash, an educational tool that provides training data for the ASL-Search dictionary[1]

My work serves as a model for how we can design accessible information resources. In particular, I presented several methodologies for designing novel character systems that can be reused to create or personalize accessible character systems in the future. These methodologies used participatory design to narrow and define design spaces (for Animated Si5s), crowdsourcing to fully explore design spaces (for Smartfonts and Livefonts), and optimization to select the best designs based on the collected data (for Smartfonts and Livefonts). The design of our ASL-to-English dictionary also serves as a model for how to design a data-dependent resource (e.g., a dictionary) for a data-scarce domain (e.g., ASL). Many domains in accessibility are under-studied and lack large-scale data needed to create powerful solutions. Our method of seamlessly aligning with crowd incentives to collect large-scale data not only shows that a volunteer crowd can be used for ASL-related tasks, but also can be applied to other data-scarce accessibility problems.

In evaluating the novel systems and designs presented in this dissertation, I also contributed new evaluation methodologies. In particular, evaluating Smartfont and Livefont legibility was difficult, as traditional legibility tests, which require participants to know how to read the script being evaluated, did not apply. We presented new evaluation methods involving matching and transcription that do not require training participants, making it newly possible to readily evaluate novel scripts, and widening the participant pool for evaluations of existing scripts. We also present new methodologies for evaluating script learnability, both in terms of immediate understandability (for Animated Si5s) and learning curve over time (for Smartfonts and Livefonts). Our simulations of the ASL-Search dictionary based on data collected through ASL-Flash also demonstrates how we can evaluate the expected performance of a system before release. Evaluating the expected performance before release can be crucial to real-world impact, as sub-par systems can be dismissed by the community, a judgement that can be difficult to rectify even with improved performance later on. These evaluation techniques can be used in future research.

---

[1] ASL-Search and ASL-Flash can also be viewed as a single larger two-part system.

8.2 Future Vision

I envision a future where systems provide accessibility for all users by intelligently adapting to each user's abilities and preferences. These systems would learn from passive interactions and explicit input, and tailor interfaces to users' abilities. Working towards this vision, I outline next steps for continuing to develop and study novel methods for both text input and display.

8.2.1 Supporting Novel Character System Use

While Smartfonts, Livefonts, and animated sign language scripts have the potential to improve the reading experience, support for installing and using these scripts does not currently exist. Few programs support custom scripts, and even fewer support scripts that are multi-colored, animated, or do more than redesign individual letters. As a result, people are forced to use default text displays for most interactions with computers, for example English (which many people do not know fluently) in a traditional font (which is difficult to read with low vision).

Developing a way to easily install custom scripts, possibly involving multicolor or animated characters, is an area for future work needed to realize the potential impact of novel character systems. Easy installation could greatly expand computer usability. Devices with default fonts that are unreadable for a variety of reasons would suddenly become correctable for many populations. For example, different scripts could be used by low-vision readers who struggle with magnification (the focus of the presented work on Smartfonts and Livefonts), and by people with dyslexia who benefit from modified text displays (e.g., [124]). In the future, scripts could also intelligently adapt to users, for example detecting selective magnification usage, and automatically adapting the display accordingly. This work feeds into a growing trend towards personalization, from tailored search results (e.g., [135]) to adaptable interfaces (e.g., [47]).

8.2.2 ASL Input

Despite demand for a usable ASL character system, little work has been done to support ASL text input. Support for both reading and writing is crucial to character system use. My work on

Animated Si5s supports reading, but support for ASL input is lacking. Generating text by signing ASL in front of a camera that identifies and transcribes signs requires conspicuous gestures and expensive hardware, and is not currently viable, as accurate sign recognition is not solved.

Supporting intuitive, efficient ASL text input is complimentary future work required for ASL character system adoption. Supporting efficient ASL input that can be used to generate ASL text has the potential for high impact on ASL users, allowing them to quickly generate and edit content in their primary language. Email clients, text editors, and search engines would become newly usable in ASL. Usage would also produce the first large-scale annotated ASL corpus. State-of-the-art sign language translation lags far behind written language translation due to a lack of large, annotated corpuses for these currently unwritten languages. Adoption of an intuitive input mechanism would produce such a corpus with only modest effort, which could be used to better train translation models.

8.2.3 Search for Content within ASL Videos

While our ASL-Search dictionary supports searching over a corpus of individual signs, our system does not support searching for content within longer videos. ASL content is primarily shared in video form, including recorded performances and lectures, and vlog and YouTube posts. Finding content in these videos is difficult and time-consuming because no support exists for search. Users currently must either watch the entire video, or view slices in an attempt to find the desired piece.

Future work to support search for content within corpuses of longer, more diverse ASL videos would greatly expand information access for ASL users. Such a search engine would make it newly possible to find desired videos or video sections within databases like YouTube or lecture series. Supporting search for content in a person's primary language is important in order to empower people to share content that can be found, and to support people finding desired information for educational, professional, and personal purposes. Just as English-based search engines like Google and Bing have made countless English resources available to English speakers, supporting search in ASL would revolutionize access to ASL-based information for the Deaf community and ASL users more generally.

8.3 Takeaways

The main takeaways from this dissertation are:

Modern computing capabilities can be leveraged to address long-standing accessibility problems. Computer scientists have developed sophisticated techniques that have been underutilized in developing accessible systems. For example, sophisticated computer graphics support rich colors, shapes, and movements. In the past, these computer graphics capabilities have not been used to improve text legibility. Instead, solutions like magnification, glasses, and customized font design limited to traditional line-drawn letterforms have been proposed. Our novel scripts (Smartfonts, Livefonts, and Animated Si5s) more fully leverage computer graphics to make text-based information significantly more accessible for low-vision readers. Our ASL-Search dictionary design also uses modern computing capabilities, specifically crowdsourcing and topic modeling, to more effectively solve the long-standing access problem of looking up signs. This work demonstrates that modern computing techniques can be leveraged to address long-standing accessibility problems in innovative ways.

Novel solutions can require breaking with tradition, which personal devices allow. Leveraging modern computing capabilities to address long-standing access problems can produce unconventional solutions. For example, by leveraging computer graphics to design character systems involving color and animation, our work on Smartfonts, Livefonts, and Animated Si5s challenges basic assumptions about the appearance of text. Text is expected to be monochromatic and stationary, as it overwhelmingly has been for thousands of years. Personal devices make breaking with tradition possible by allowing individuals to adopt new solutions (in this case, character systems) without impacting anybody else's experience. Similarly, our ASL-Search dictionary and ASL-Flash flashcards provide benefit to those who choose to use them without burdening others. Unconventional systems need not be adopted by everybody for people to use them and derive benefit.

Novel solutions can be designed in principled ways using data. Each system presented in this dissertation was the result of data-driven design. To design Smartfonts and Livefonts expected

to improve legibility, we optimized over crowdsourced perceptual data. In designing our Livefonts and Animated Si5s, we also gathered data from target users through participatory design procedures to narrow and define our design spaces. Our ASL-Search dictionary design is completely powered by data collected from ASL users, crowdsourced through the complementary ASL-Flash site and collected from dictionary users themselves. The effectiveness of the dictionary design is derived from the fact that it is based on data collected from users, which allows for a more robust language model than previous dictionary designs. Grounding design decisions in data supports the creation of effective accessibility solutions.

BIBLIOGRAPHY

[1] Aisha Shamsuna Ahmed and Daniel Su Kuen Seong. SignWriting on Mobile Phones for the Deaf. In *Proc. of Mobility*. ACM, 2006.

[2] M Alpern. Accommodation and the Pupil. Part II. *The Eye*, 3, 1969.

[3] Ira B Appelman and Mark S Mayzner. Application of geometric models to letter recognition: Distance and density. *Journal of Experimental Psychology: General*, 111(1):60, 1982.

[4] Aries Arditi. Adjustable typography: an approach to enhancing low vision text accessibility. *Ergonomics*, 47(5):469–482, 2004.

[5] Aries Arditi and Jianna Cho. Serifs and font legibility. *Vision Research*, 45(23):2926–2933, 2005.

[6] Rudolf Arnheim. *Art and visual perception: A psychology of the creative eye*. Univ of California Press, 1956.

[7] F Gregory Ashby and Nancy A Perrin. Toward a unified theory of similarity and recognition. *Psychological Review*, 95(1):124, 1988.

[8] ASLized! The Official American Sign Language Writing Font by ASLized & si5s. `http://aslized.org/font-download/`, 2017. (Accessed 2018-03-02).

[9] ASLPro, May 2014. `http://www.aslpro.com`.

[10] RA Augustus, E Ritchie, and S Stecker. The Official American Sign Language Writing Textbook. *ASLized*, 2013.

[11] Ian L Bailey and JAN E LOVIE. The design and use of a new near-vision chart. *Optometry and Vision Science*, 57(6):378–387, 1980.

[12] Jean-Baptiste Bernard, Girish Kumar, Jasmine Junge, and Susana TL Chung. The effect of letter-stroke boldness on reading speed in central and peripheral vision. *Vision Research*, 84:33–42, 2013.

[13] Michael Bernard, Chia Hui Liao, and Melissa Mills. The effects of font type and size on the legibility and reading time of online text by older adults. In *Ext. Abstracts CHI 2001*, pages 175–176. ACM Press, 2001.

[14] Michael L Bernard, Barbara S Chaparro, Melissa M Mills, and Charles G Halcomb. Comparing the effects of text size and format on the readibility of computer-displayed Times New Roman and Arial text.

[15] Ann Bessemans. Matilda: a typeface for children with low vision. *Digital Fonts and Reading*, 1:19, 2016.

[16] Ann Bessemans et al. *Letterontwerp voor kinderen met een visuele functiebeperking*. PhD thesis, Leiden University, 2012.

[17] Jeffrey P Bigham, Daniel S Otero, Jessica N DeWitt, Anna Cavender, and Richard E Ladner. ASL-STEM Forum: A Bottom-Up Approach to Enabling American Sign Language to Grow in STEM Fields. In *ICWSM*, 2008.

[18] Frans JJ Blommaert and Han Timmers. Letter recognition at low contrast levels: effects of letter size. *Perception*, 16(4):421–432, 1987.

[19] Susanne Bødker, Pelle Ehn, Dan Sjögren, and Yngve Sundblad. Co-operative Design–Perspectives on 20 Years with 'the Scandinavian IT Design Model'. In *Proc. of NordiCHI*, 2000.

[20] Rick Bonney, Caren B Cooper, Janis Dickinson, Steve Kelling, Tina Phillips, Kenneth V Rosenberg, and Jennifer Shirk. Citizen science: a developing tool for expanding science knowledge and scientific literacy. *BioScience*, 59(11):977–984, 2009.

[21] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *IEEE TVCG*, 17(12):2301–2309, 2011.

[22] Danielle Bragg, Shiri Azenkot, and Adam Tauman Kalai. Reading and Learning Smartfonts. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 391–402. ACM, 2016.

[23] Danielle Bragg, Shiri Azenkot, Kevin Larson, Ann Bessemans, and Adam Tauman Kalai. Designing and Evaluating Livefonts. In *Proceedings of the 30th Annual Symposium on User Interface Software and Technology*, pages 481–492. ACM, 2017.

[24] Danielle Bragg, Raja Kushalnagar, and Richard Ladner. Designing an Animated Character System for American Sign Language. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 2018. To appear.

[25] Danielle Bragg, Kyle Rector, and Richard E Ladner. A User-Powered American Sign Language Dictionary. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1837–1848. ACM, 2015.

[26] Matthew W Brault et al. *Americans with Disabilities: 2010*. US Department of Commerce, Economics and Statistics Administration, US Census Bureau, 2012.

[27] Mary Brennan, Martin D Colville, Lilian K Lawson, and Gerry Hughes. *Words in hand: A structural analysis of the signs of British Sign Language*. Edinburgh BSL Research Project, 1984.

[28] Justin Cheng, Jaime Teevan, and Michael S Bernstein. Measuring crowdsourcing effort with error-time curves. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1365–1374. ACM, 2015.

[29] Olympia Colizoli, Jaap MJ Murre, and Romke Rouw. Pseudo-synesthesia through reading books with colored letters. *PLoS One*, 7(6):e39799, 2012.

[30] Max Coltheart, Kathleen Rastle, Conrad Perry, Robyn Langdon, and Johannes Ziegler. DRC: a dual route cascaded model of visual word recognition and reading aloud. *Psychological review*, 108(1):204, 2001.

[31] The Unicode Consortium. The Unicode Standard, Version 8.0. `http://www.unicode.org/versions/Unicode8.0.0/`, 2015.

[32] Helen Cooper, Nicolas Pugeault, and Richard Bowden. Reading the signs: A video based sign dictionary. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 914–919. IEEE, 2011.

[33] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

[34] Antonio Carlos da Rocha Costa and Graçaliz Pereira Dimuro. SignWriting and SWML–Paving the way to Sign Language Processing. *Atelier Traitement Automatique des Langues des Signes, TALN 2003*, 2003.

[35] Karin Danielsson, Amir M Naghsh, Dorina Gumm, and Andrew Warr. Distributed Participatory Design. In *CHI'08 extended abstracts on Human factors in computing systems*, pages 3953–3956. ACM, 2008.

[36] Iain Darroch, Joy Goodman, Stephen Brewster, and Phil Gray. The effect of age and font size on reading text on handheld computers. In *Proc. INTERACT 2005*, pages 253–266. Springer, 2005.

[37] Joseph Davis and Elsie Stecker. Deaf-Blind Ninja. ASLized! Journal of American Sign Language and Literature, 2011. Retrieved from `http://aslized.org/journal/ninja`.

[38] M Davis and P Edberg. Unicode emoji-unicode technical report# 51. Technical report, Technical Report 51 (3), 2016.

[39] Matjaž Debevc, Primož Kosec, and Andreas Holzinger. Improving Multimodal Web Accessibility for Deaf People: Sign Language Interpreter Module. *Multimedia Tools and Applications*, 54(1):181–199, 2011.

[40] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the Association for Information Science and Technology (JASIS)*, 41(6):391–407, 1990.

[41] Cagatay Demiralp Demiralp, Michael S Bernstein, and Jeffrey Heer. Learning Perceptual Kernels for Visualization Design. *IEEE TVCG*, 20(12):1933–1942, 2014.

[42] Jonathan Dobres, Bryan Reimer, Bruce Mehler, Nadine Chahine, and David Gould. A Pilot Study Measuring the Relative Legibility of Five Simplified Chinese Typefaces Using Psychophysical Methods. In *Proc. AutomotiveUI 2014*, pages 1–5. ACM, 2014.

[43] Alexander Duane. Normal values of the accommodation at all ages. *Journal of the American Medical Association*, 59(12):1010–1013, 1912.

[44] Eleni Efthimiou and Stavroula-Evita Fotinea. An environment for deaf accessibility to educational content. *Greek national project DIANOEMA,(GSRT, M3. 3, id 35), Hammamet-Tunisia, ICTA*, 2007.

[45] Ralph Elliott, Helen Cooper, Eng-Jon Ong, John Glauert, Richard Bowden, and F Lefebvre-Albaret. Search-by-example in multilingual sign language databases. In *Proc. of Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT)*, 2011.

[46] Ralph Elliott, JRW Glauert, Vince Jennings, and JR Kennaway. An Overview of the SiGML Notation and SiGMLSigning Software System. In *Sign Language Processing Satellite Workshop of the Fourth International Conference on Language Resources and Evaluation, LREC 2004*, pages 98–104, 2004.

[47] Leah Findlater and Joanna McGrenere. A Comparison of Static, Adaptive, and Adaptable Menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 89–96. ACM, 2004.

[48] Donald C Fletcher, Ronald A Schuchard, and Gale Watson. Relative locations of macular scotomas near the PRL: effect on low vision reading. *Journal of Rehabilitation Research and Development*, 36(4):356–364, 1999.

[49] Christiane Floyd, Wolf-Michael Mehl, Fanny-Michaela Reisin, Gerhard Schmidt, and Gregor Wolf. Out of Scandinavia: Alternative Approaches to Software Design and System Development. *Human-Computer Interaction*, 4(4):253–350, 1989.

[50] American Printing House for the Blind Inc. APHont™: A Font for Low Vision. `http://www.aph.org/products/aphont/`, 2004. (Accessed 2017-03-12).

[51] Christopher Frauenberger, Judith Good, and Wendy Keay-Bright. Designing Technology for Children with Special Needs: Bridging Perspectives through Participatory Design. *CoDesign*, 7(1):1–28, 2011.

[52] Russell Freedman. *Out of darkness: the story of Louis Braille*.

[53] Marelys L Garcia and Cesar I Caldera. The effect of color and typeface on the readability of on-line text. *Computers and Industrial Engineering*, 31(1):519–524, 1996.

[54] Wilson Geisler and Richard Murray. Cognitive neuroscience: Practice doesn't make perfect. *Nature*, 423(6941):696–697, 2003.

[55] Laura Germine, Ken Nakayama, Bradley C Duchaine, Christopher F Chabris, Garga Chatterjee, and Jeremy B Wilmer. Is the Web as good as the lab? Comparable performance from Web and lab in cognitive/perceptual experiments. *Psychonomic bulletin & review*, 19(5):847–857, 2012.

[56] Dafydd Gibbon, Ulrike Gut, Benjamin Hell, Karin Looks, Alexandra Thies, and Thorsten Trippel. A Computational Model of Arm Gestures in Conversation. In *INTERSPEECH*, 2003.

[57] John Gill and Sylvie Perera. Accessible universal design of interactive digital television. In *Proceedings of the 1st European conference on interactive television: from viewers to actors*, pages 83–89. Citeseer, 2003.

[58] Paul Green-Armytage. A colour alphabet and the limits of colour coding.

[59] Donald A Grushkin. Writing Signed Languages: What For? What Form? *American Annals of the Deaf*, 161(5):509–527, 2017.

[60] Dorina C Gumm, Monique Janneck, and Matthias Finck. Distributed Participatory Design–a Case Study. In *Proc. of the DPD Workshop at NordiCHI*, volume 2, 2006.

[61] Richard H Hall and Patrick Hanna. The impact of web page text-background colour combinations on readability, retention, aesthetics and behavioural intention. *Behaviour and Information Technology*, 23(3):183–195, 2004.

[62] Thomas Hanke. iLex-A tool for Sign Language Lexicography and Corpus Analysis. In *LREC*, 2002.

[63] Thomas Hanke. HamNoSys-Representing Sign Language Data in Language Resources and Language Processing Contexts. In *LREC*, volume 4, 2004.

[64] Vicki L Hanson and Carol A Padden. Interactive video for bilingual ASL/English instruction of deaf children. *American Annals of the Deaf*, 134(3):209–213, 1989.

[65] Jeffrey Heer and Michael Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proc. CHI 2010*, pages 203–212. ACM, 2010.

[66] Frank G Hofmann and Günter Hommel. Analyzing human gestural motions using acceleration sensors. In *Proc. of Gesture Workshop on Progress in Gestural Interaction*, pages 39–59. Springer-Verlag, 1996.

[67] Judith A Holt. Stanford Achievement Test-8th edition: Reading Comprehension Subgroup Results. *American Annals of the Deaf*, 138(2):172–175, 1993.

[68] Weiyin Hong, James YL Thong, and Kar Yan Tam. Does animation attract online usersâĂŹ attention? The effects of flash on information search performance and perceptions. *Information Systems Research*, 15(1):60–86, 2004.

[69] John J Horton, David G Rand, and Richard J Zeckhauser. The online laboratory: Conducting experiments in a real labor market. *Experimental economics*, 14(3):399–425, 2011.

[70] Matt Huenerfauth. A Multi-Path Architecture for Machine Translation of English Text into American Sign Language Animation. In *Proc. of the Student Research Workshop at HLT-NAACL*, pages 25–30. Association for Computational Linguistics, 2004.

[71] Matt Huenerfauth. *Generating American Sign Language Classifier Predicates for English-to-ASL Machine Translation*. PhD thesis, University of Pennsylvania, 2006.

[72] Matt Huenerfauth and Vicki L Hanson. Sign Language in the Interface. In*The Universal Access Handbook*, pages 1–18. CRC Press, 2009.

[73] Kazuyuki Imagawa, Shan Lu, and Seiji Igi. Color-based hands tracking system for sign language recognition. In *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 462–467. IEEE, 1998.

[74] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[75] Richard M Karp. Reducibility among Combinatorial Problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[76] Ryan Kelly and Leon Watts. Characterising the inventive appropriation of emoji as relationally meaningful in mediated close personal relationships. *Experiences of Technology Appropriation: Unanticipated Users, Usage, Circumstances, and Design*, 2015.

[77] Adam Kendon. *Sign Languages of Aboriginal Australia: Cultural, Semiotic and Communicative Perspectives*. Cambridge University Press, 1988.

[78] JR Kennaway, John RW Glauert, and Inge Zwitserlood. Providing signed content on the Internet by synthesized animation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(3):15, 2007.

[79] Finn Kensing and Jeanette Blomberg. Participatory Design: Issues and Concerns. *Computer Supported Cooperative Work (CSCW)*, 7(3-4):167–185, 1998.

[80] Ji-Hwan Kim, Nguyen Duc Thang, and Tae-Seong Kim. 3-D hand motion tracking and gesture recognition using a data glove. In *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on*, pages 1013–1018. IEEE, 2009.

[81] Jong-Sung Kim, Won Jang, and Zeungnam Bien. A dynamic gesture recognition system for the Korean Sign Language (KSL). *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(2):354–359, 1996.

[82] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, 2015.

[83] Stefan Kopp, Timo Sowa, and Ipke Wachsmuth. Imitation games with an artificial agent: From mimicking to understanding shape-related iconic gestures. In *International Gesture Workshop*, pages 436–447. Springer, 2003.

[84] MiYoung Kwon and Gordon E Legge. Higher-contrast requirements for recognizing low-pass–filtered letters. *Journal of Vision*, 13(1):13–13, 2013.

[85] Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[86] Joseph S Lappin, Duje Tadin, Jeffrey B Nyquist, and Anne L Corn. Spatial and temporal limits of motion perception across variations in speed, eccentricity, and low vision. *Journal of Vision*, 9(1):30–30, 2009.

[87] Gordon E Legge. Psychophysics of reading in normal and low vision. In *OSA Noninvasive Assessment of the Visual System, 1993, Monterey; Portions of this research (MNREAD acuity charts) were presented at the aforementioned conference.* Lawrence Erlbaum Associates Publishers, 2007.

[88] Gordon E Legge and Charles A Bigelow. Does print size matter for reading? A review of findings from vision science and typography. *Journal of Vision*, 11(5):8, 2011.

[89] Gordon E Legge, J Stephen Mansfield, and Susana TL Chung. Psychophysics of reading: XX. Linking letter recognition to reading speed in central and peripheral vision. *Vision Research*, 41(6):725–743, 2001.

[90] Gordon E Legge, David H Parish, Andrew Luebker, and Lee H Wurm. Psychophysics of reading: XI. Comparing color contrast and luminance contrast. *Journal of the Optical Society of America A*, 7(10):2002–2010, 1990.

[91] Gordon E Legge, Denis G Pelli, Gar S Rubin, and Mary M Schleske. Psychophysics of reading - I. Normal vision. *Vision research*, 25(2):239–252, 1985.

[92] Gordon E Legge, Julie A Ross, Lisa M Isenberg, and James M Lamay. Psychophysics of reading. Clinical predictors of low-vision reading speed. *Investigative Ophthalmology & Visual Science*, 33(3):677–687, 1992.

[93] Gordon E Legge and Gary S Rubin. Psychophysics of reading. IV. Wavelength effects in normal and low vision. *JOSA A*, 3(1):40–51, 1986.

[94] Gordon E Legge, Gary S Rubin, and Andrew Luebker. Psychophysics of reading-V. The role of contrast in normal vision. *Vision research*, 27(7):1165–1177, 1987.

[95] Gordon E Legge, Gary S Rubin, and Andrew Luebker. Psychophysics of reading: V. The role of contrast in normal vision. *Vision Research*, 27(7):1165–1177, 1987.

[96] Ella Mae Lentz, Ken Mikos, and Cheri Smith. *Signing Naturally: Student Workbook Units 1-6*. Dawn Sign Press, 2008.

[97] Ella Mae Lentz, Ken Mikos, and Cheri Smith. *Signing Naturally: Student Workbook Units 7-12*. Dawn Sign Press, 2014.

[98] Rung-Huei Liang and Ming Ouhyoung. A real-time continuous gesture recognition system for sign language. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 558–567. IEEE, 1998.

[99] Stephen Lindsay, Katie Brittain, Daniel Jackson, Cassim Ladha, Karim Ladha, and Patrick Olivier. Empathy, Participatory Design and People with Dementia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 521–530. ACM, 2012.

[100] Jack M Loomis. On the tangibility of letters and braille. *Perception and Psychophysics*, 29(1):37–46, 1981.

[101] Jack M Loomis. A model of character recognition and legibility. *Journal of Experimental Psychology: Human Perception and Performance*, 16(1):106, 1990.

[102] JS Mansfield, SJ Ahn, GE Legge, and A Luebker. A new reading-acuity chart for normal and low vision. *Ophthalmic and Visual Optics/Noninvasive Assessment of the Visual System Technical Digest*, 3:232–235, 1993.

[103] Craig H Martell. An extensible, kinematically-based gesture annotation scheme. 2005.

[104] C.P. Masica. *The Indo-Aryan Languages*. Cambridge Language Surveys. Cambridge University Press, 1993.

[105] Marina McIntire, Don Newkirk, Sandra Hutchins, and Howard Poizner. Hands and faces: A preliminary inventory for written ASL. *Sign Language Studies*, 56(1):197–241, 1987.

[106] Chris McKinstry, Rick Dale, and Michael J Spivey. Action dynamics reveal parallel competition in decision making. *Psychological Science*, 19(1):22–24, 2008.

[107] Christopher Miller. Section I: Some reflections on the need for a common sign notation. *Sign Language & Linguistics*, 4(1):11–28, 2001.

[108] Modern Language Association (MLA). Enrollments in Languages Other than English in United States Institutions of Higher Education, Summer 2016 and Fall 2016: Preliminary Report. `http://www.mla.org/content/download/83540/2197676/2016-Enrollments-Short-Report.pdf`, 2018. (Accessed: 2018-03-04).

[109] Sara Morrissey and Andy Way. An Example-Based Approach to Translating Sign Language. *Workshop in Example-Based Machine Translation (MT Summit X)*, pages 109–116, 2005.

[110] Robert A Moses. Accommodation. *Adler's Physiology of the Eye*, pages 291–310, 1987.

[111] Mayumi Negishi. Meet Shigetaka Kurita, the Father of Emoji. *Wall Street Journal*, 2014.

[112] Don Newkirk. *SignFont Handbook*. San Diego: Emerson and Associates, 1987.

[113] World Federation of the Deaf. Know and Achieve your Human Rights Toolkit. `http://wfdeaf.org/our-work/human-rights-of-the-deaf/`, 2016.

[114] Eng-Jon Ong, Helen Cooper, Nicolas Pugeault, and Richard Bowden. Sign language recognition using sequential pattern trees. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2200–2207. IEEE, 2012.

[115] World Health Organization. Vision Impairment and Blindness: Fact Sheet. `http://www.who.int/mediacentre/factsheets/fs282/en/`, 2017.

[116] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. Running experiments on amazon mechanical turk. 2010.

[117] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[118] Denis G Pelli, Catherine W Burns, Bart Farell, and Deborah C Moore-Page. Feature detection and letter identification. *Vision Research*, 46(28):4646–4674, 2006.

[119] David Peterson. Sign Language International Phonetic Alphabet (SLIPA). `http://dedalvs.conlang.org/slipa.html`, 2013. (Accessed 2017-05-07).

[120] Helen Petrie, Wendy Fisher, Kurt Weimann, and Gerhard Weber. Augmenting Icons for Deaf Computer Users. In *CHI'04 Extended Abstracts on Human Factors in Computing Systems*, pages 1131–1134. ACM, 2004.

[121] Christopher Plaue and John Stasko. Animation in a peripheral display: distraction, appeal, and information conveyance in varying display configurations. In *Proceedings of Graphics Interface 2007*, pages 135–142. ACM, 2007.

[122] D Purves, GJ Augustine, D Fitzpatrick, LC Katz, AS LaMantia, JO McNamara, et al. Anatomical Distribution of Rods and Cones. *Neuroscience. Sunderland (MA): Sinauer Associates*, 2001.

[123] Katharina Reinecke and Krzysztof Z Gajos. LabintheWild: Conducting large-scale online experiments with uncompensated samples. In *CSCW '15*, pages 1364–1378. ACM, 2015.

[124] Luz Rello and Ricardo Baeza-Yates. Good fonts for dyslexia. In *Proc. ASSETS 2013*, page 14. ACM, 2013.

[125] Interagency Language Roundtable. Interagency Language Roundtable (ILR) Scale, 1985.

[126] Yong Rui, Thomas S Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10(1):39–62, 1999.

[127] Ruby Ryles. The impact of braille reading skills on employment, income, education, and reading habits. *Journal of Visual Impairment and Blindness*, 90:219–226, 1996.

[128] Signing Savvy. ASL Sign Language Video Dictionary. `http://www.signingsavvy.com`, 2018. (Accessed 2018-04-05).

[129] Douglas Schuler and Aki Namioka. *Participatory Design: Principles and Practices*. CRC Press, 1993.

[130] Kong-King Shieh and Chin-Chiuan Lin. Effects of screen type, ambient illumination, and color combination on VDT visual performance and subjective preference. *Intl. Journal of Industrial Ergonomics*, 26(5):527–536, 2000.

[131] Jonathan Silvertown. A new dawn for citizen science. *Trends in ecology & evolution*, 24(9):467–471, 2009.

[132] SignWriting Site. SignWriting in Brazil. `http://www.signwriting.org/brazil/`, 2017.

[133] Louise L Sloan. Reading aids for the partially sighted. *American Journal of Optometry and Physiological Optics*, 54(9):646, 1977.

[134] Frank Smith. *Understanding reading: A psycholinguistic analysis of reading and learning to read*. Routledge, 2012.

[135] Mirco Speretta and Susan Gauch. Personalized Search Based on User Search Histories. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 622–628. IEEE, 2005.

[136] Ai Squared. ZoomText. http://www.aisquared.com/products/zoomtext/, 2017. (Accessed 2017-03-13).

[137] Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden Markov models. In *Motion-Based Recognition*, pages 227–243. Springer, 1997.

[138] William C Stokoe. Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf. *Studies in Linguistics: Occasional Papers*, 8, 1960.

[139] William C Stokoe. Sign language structure. 1978.

[140] William C Stokoe, Dorothy C Casterline, and Carl G Croneberg. *A Dictionary of American Sign Language on Linguistic Principles*. Linstok Press Silver Spring, 1965 (reissued 1976).

[141] Thomas Stone. ASLSJ. https://sites.google.com/site/aslsignjotting/, 2009. (Accessed 2017-04-05).

[142] Katja Straetz, Andreas Kaibel, Vivian Raithel, Marcus Specht, Klaudia Grote, and Florian Kramer. An E-Learning Environment for Deaf Adults. In *Proc. 8th ERCIM workshop*, 2004.

[143] Samuel Supalla. ASL-phabet. http://www.aslphabet.com/home.html, 2012. (Accessed 2017-05-07).

[144] Samuel J Supalla, Jody H Cripps, and Andrew PJ Byrne. Why American Sign Language Gloss Must Matter. *American Annals of the Deaf*, 161(5):540–551, 2017.

[145] Samuel J Supalla, Tina R Wix, and Cecile McKee. Print as a primary source of English for deaf learners. *One mind, two languages: Studies in bilingual language processing*, pages 177–190, 2001.

[146] Valerie Sutton. *Lessons in Sign Writing: Textbook*. SignWriting, 1995.

[147] Sarit Felicia Anais Szpiro, Shafeka Hashash, Yuhang Zhao, and Shiri Azenkot. How People with Low Vision Access Computing Devices: Understanding Challenges and Opportunities. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 171–180. ACM, 2016.

[148] Duje Tadin, Jeffrey B Nyquist, Kelly E Lusk, Anne L Corn, and Joseph S Lappin. Peripheral Vision of Youths with Low Vision: Motion Perception, Crowding, and Visual Search Peripheral Function in Youths with Low Vision. *Investigative ophthalmology & visual science*, 53(9):5860–5868, 2012.

[149] TEDx Talks. Robert Arnold Augustus at TEDxIslay. https://www.youtube.com/watch?v=SYHs3D6jJ_o&t=271s, 2012. (Accessed 2017-05-04).

[150] Richard A Tennant and Marianne Gluszak Brown. *The American Sign Language Handshape Dictionary*. Gallaudet University Press, 1998.

[151] Michael Toomim, Travis Kriplean, Claus Pörtner, and James Landay. Utility of human-computer interactions: Toward a science of preference measurement. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2275–2284. ACM, 2011.

[152] James T Townsend. A note on the identifiability of parallel and serial processes. *Perception & Psychophysics*, 10(3):161–163, 1971.

[153] Matthew J Traxler. *Introduction to psycholinguistics: Understanding language science*. John Wiley & Sons, 2011.

[154] The Ultimate American Sign Language Dictionary, May 2013. http://idrt.com/store/the-ultimate-american-sign-language-dictionary.

[155] Gerard Unger. *While you're reading*. Mark Batty Pub, 2007.

[156] Gallaudet University. PST 275 - Introduction to ASL Writing: Si5s System. http://www.gallaudet.edu/academic-catalog/center-for-continuing-studies/pst-courses, 2017-2018. (Accessed 2018-03-02).

[157] Jean-Luc Vinot and Sylvie Athènes. Legible, are you sure?: an experimentation-based typographical design in safety-critical context. In *Proc. CHI 2012*, pages 2287–2296. ACM, 2012.

[158] Luis von Ahn. Duolingo: learn a language for free while helping to translate the web. In *Proceedings of the 2013 international conference on Intelligent user interfaces (IUI)*, pages 1–2. ACM Press, 2013.

[159] Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In *CHI '04*, pages 319–326. ACM, 2004.

[160] Haijing Wang, Alexandra Stefan, Sajjad Moradi, Vassilis Athitsos, Carol Neidle, and Farhad Kamangar. A system for large vocabulary sign search. In *Trends and Topics in Computer Vision*, pages 342–353. Springer, 2012.

[161] Lujin Wang, Joachim Giesen, Kevin T McDonnell, Peter Zolliker, and Klaus Mueller. Color design for illustrative visualization. *IEEE TVCG*, 14(6):1739–1754, 2008.

[162] Robert Y Wang and Jovan Popović. Real-time hand-tracking with a color glove. In *ACM Transactions on Graphics (TOG)*, volume 28, page 63. ACM, 2009.

[163] Stephen G Whittaker and JAN Lovie-Kitchin. Visual requirements for reading. *Optometry & Vision Science*, 70(1):54–65, 1993.

[164] Alexandra Wiebelt. Do symmetrical letter pairs affect readability?: A cross-linguistic examination of writing systems with specific reference to the runes. *Written Language and Literacy*, 7(2):275–303, 2004.

[165] Ronnie B Wilbur. Modality and the structure of language: Sign languages versus signed systems. *Oxford handbook of deaf studies, language, and education*, pages 332–346, 2003.

[166] Arnold J Wilkins, Nirmal Sihra, and Andrew Myers. Increasing reading speed by using colours: issues concerning reliability and specificity, and their theoretical and practical implications. *Perception*, 34(1):109–120, 2005.

[167] Jonathan Winawer, Nathan Witthoft, Michael C Frank, Lisa Wu, Alex R Wade, and Lera Boroditsky. Russian blues reveal effects of language on color discrimination. *Proceedings of the National Academy of Sciences*, 104(19):7780–7785, 2007.

[168] World Bank Group World Health Organization. World Report on Disability. `http://www.who.int/disabilities/world_report/2011/report/en/`, 2011.

[169] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[170] Ming-Hsuan Yang and Narendra Ahuja. Recognizing hand gestures using motion trajectories. In *Face Detection and Gesture Recognition for Human-Computer Interaction*, pages 53–81. Springer, 2001.

[171] Zahoor Zafrulla, Helene Brashear, Thad Starner, Harley Hamilton, and Peter Presti. American Sign Language Recognition with the Kinect. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 279–286. ACM, 2011.

[172] Norziha Megat Mohd Zainuddin, Halimah Badioze Zaman, and Azlina Ahmad. A Participatory Design in Developing Prototype an Augmented Reality Book for Deaf Students. In *2nd International Conference on Computer Research and Development*. IEEE, 2010.

[173] Liwei Zhao, Karin Kipper, William Schuler, Christian Vogler, Norman Badler, and Martha Palmer. A Machine Translation System from English to American Sign Language. *Envisioning Machine Translation in the Information Future*, pages 191–193, 2000.

[174] Yuhang Zhao, Sarit Szpiro, and Shiri Azenkot. Foresee: A customizable head-mounted vision enhancement system for people with low vision. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, pages 239–249. ACM, 2015.

[175] Jason E Zinza. *Master ASL!* Sign Media Inc, 2006.

[176] Jason E Zinza. *Master ASL!: Level One*. Sign Media Inc., 2006.

Appendix A

SUPPLEMENTARY MATERIALS

A.1 Proof: Selecting k nodes of a graph with minimal sum of edge weights is NP-hard

Problem: Given a weighted graph $G = (V,E)$ where $E = \{e_{i,j} \in [0,1] | i,j \in V\}$ and positive integer k, find set $S \subseteq V$, $|S| = k$ so as to minimize $\sum_{i,j \in S, i \neq j} e_{ij}$.

This problem is NP-Hard, by the following reduction from the Clique Problem (determining whether or not a clique of a certain size exists in a given graph).

Algorithm A (black box for our problem):

Input: positive integer k, weighted graph $G = (E,V)$ where $E = \{e_{i,j} \in [0,1] | i,j \in V\}$

Output: $S \subseteq V$

Algorithm B:

Input: positive integer k, unweighted graph $G = (V,E)$, $E = \{e_{i,j} \in \{0,1\} | i,j \in V\}$

Create graph G^I by flipping the edges of G: $G^I = (V,E^I)$, $E^I = \{1 - e_{i,j} | i,j \in V\}$.

Run Algorithm A on input k, G^I, to return S_I.

If $\sum_{i,j \in S_I} e_{i,j} = 0$, return True.

Else, return False.

As Algorithm B is a polynomial time reduction from our problem (solved by Algorithm A) to the Clique Problem, and the Clique Problem is NP-hard [75], our problem is NP-hard.

A.2 Proof: Partitioning a graph into three sets of nodes with minimal sum of edge weights within sets is NP-hard

Problem: Given a weighted graph $G = (V,E)$ where $E = \{e_{i,j} \in [0,1] | i,j \in V\}$, partition the nodes into 3 sets $S = S_1 \cup S_2 \cup S_3$ so as to minimize $\sum_{k=1}^{3} \sum_{i,j \in S_k, i \neq j} e_{i,j}$

This problem is NP-Hard, by the following reduction from the Max-Cut Problem (determining a graph cut that maximizes the sum of the cut edges).

Algorithm A (black box for our problem):

Input: weighted graph $G = (V, E)$, $E = \{e_{i,j} \in [0,1] | i, j \in V\}$

Output: S_1, S_2, S_3, a partition of V.

Algorithm B:

Input: weighted graph $G = (V, E)$, $E = \{e_{i,j} \in [0,1] | i, j \in V\}$

Create graph G^I by adding a set of nodes W, $|W| = |V|$ to G, with edges $\{e_{i,j} = 1 | i \in W, j \in V\}$, and $\{e_{i,j} = 0 | i, j \in W\}$.

Run Algorithm A on input G^I, to return S_1, S_2, S_3. It can be shown that if G is not the empty graph (which is trivial to handle), then one of S_1, S_2, S_3 must be exactly W. Without loss of generality, $S_3 = W$.

Return S_1, S_2.

As Algorithm B is a polynomial time reduction from our problem (solved by Algorithm A) to the Max-Cut Problem, and the Max-Cut Problem is NP-hard [75], our problem is NP-hard.

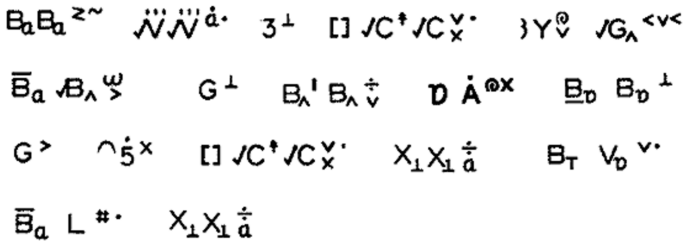
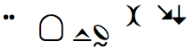
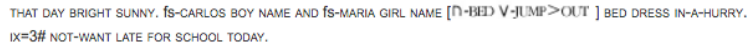
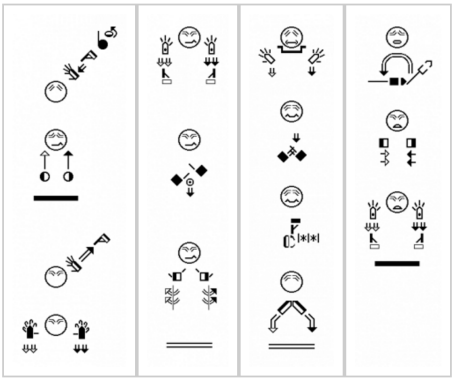
Name	Caption	Example
Stokoe Notation	Goldilocks excerpt.[1]	
HamNoSys	"House" [2]	
Gloss	Excerpt about children going to school.[3]	
SignWriting	Dr. Suess *The Cat in the Hat.* excerpt.[4]	
Si5s	"You go to school tomorrow."[5]	

Table A.1: Examples of popular ASL writing systems.

[1] Source: https://upload.wikimedia.org/wikipedia/commons/d/d3/Stokoe_passage.gif

[2] Source: [63]

[3] Source: [144]

[4] Source: Cherie Wren, http://www.signbank.org/signpuddle2.0/canvas.php?ui=1&sgn=5&sid=144

[5] Source: ASLized, http://aslized.org/files/ASLwritingslides.pdf