FatFonts:
Combining the Symbolic and Visual Aspects of Numbers

Miguel Nacenta
School of Computer Science
SACHI Group
University of St Andrews, UK
mans@st-andrews.ac.uk

Uta Hinrichs
Dept. of Computer Science
InnoVis Group
University of Calgary, Canada
uhinrich@ucalgary.ca

Sheelagh Carpendale
Dept. of Computer Science
InnoVis Group
University of Calgary, Canada
sheelagh@ucalgary.ca

ABSTRACT

In this paper we explore numeric typeface design for visualization purposes. We introduce FatFonts, a technique for visualizing quantitative data that bridges the gap between numeric and visual representations. FatFonts are based on Arabic numerals but, unlike regular numeric typefaces, the amount of ink (dark pixels) used for each digit is proportional to its quantitative value. This enables accurate reading of the numerical data while preserving an overall visual context. We discuss the challenges of this approach that we identified through our design process and propose a set of design goals that include legibility, familiarity, readability, spatial precision, dynamic range, and resolution. We contribute four FatFont typefaces that are derived from our exploration of the design space that these goals introduce. Finally, we discuss three example scenarios that show how FatFonts can be used for visualization purposes as valuable representation alternatives.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: Misc.

General Terms

Design

Keywords

Information Visualization, Typography, Numerals

1. INTRODUCTION

In this paper we introduce the concept, design and software implementation of FatFonts, a visualization technique that integrates the symbolic and visual aspects of numbers. A numeral system is "a writing system for expressing numbers" [27]. Although generally taken for granted, current numeral systems are sophisticated tools that result from millennia of development. For instance, the Hindu-Arabic numeral system, that we use every day, is based on ten separate symbols to represent the first ten numbers in base-10

(including zero), and utilizes a positional system to allow the representation of higher orders of magnitude.

Numeral systems have not, however, always been so flexible and sophisticated; more simple systems exist. In the *unary numeral system* (see Fig. 1(a), top), for instance, each counted object is represented by a mark. Unary numeral systems are still used by, e.g., bartenders or prisoners to count beers on credit and days spent in jail, but they have the obvious disadvantage that large numbers become difficult to identify and manipulate. Fig. 1(a) shows how representing the number 34 (still relatively small) in the unary system is tedious to write, difficult to read, and space consuming. This might be why the Babylonians and many other civilizations developed more abstract symbols that support the representation of large numbers without correspondingly larger numeral combinations (see Fig. 1(a)).
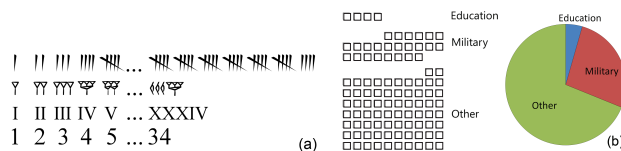

Figure 1: (a) The unary, Babylonian, Roman, and Hindu-Arabic numerals (top to bottom). (b) U.S. education and military spending represented by blocks and pie sectors.

However, abstraction also removes the direct relationship between the visual and the quantitative: we do not anymore *see* a certain quantity, but have to learn to *interpret* a symbol that represents the quantity. In fact, the discipline of information visualization [25] is often concerned with providing visual representations that facilitate the comparison, perception, and comprehension of quantities. For example, Fig. 1(b), shows two simple visualizations of military and education spending in the 2011 U.S. federal budget. Both visualizations can be regarded as numbers expressed in a unary system: the visualization on the left uses rectangles as the basic unit, and the visualization on the right uses area. However, such visualizations suffer from similar problems as the unary numerals in Fig. 1(a): it is difficult to identify exact numbers and operate with them.

This paper describes our explorations in combining the symbolic and the visual aspects of numbers in a single, hybrid technique: FatFonts (see Fig. 2). FatFonts are glyphs shaped according to Hindu-Arabic numerals (the symbolic aspect) that cover an area proportional to the represented

Figure 2: A scalar field represented with FatFonts.

value (the visual aspect). In other words, FatFonts are numerals that retain the recognizable shape of Hindu-Arabic numerals while their area (or amount of 'ink') is proportional to the represented number: number two's ink area is double number one's, number three's is triple number one's, and so on. In this way, FatFonts provide simultaneously a symbolic and visual quantitative reading of data, contributing to the set of existing visual representation techniques in static, dynamic, and interactive information systems. FatFonts lend themselves to the visualization of large data sets where both overview and numerical detail of data is important. The following sections describe the FatFonts technique, our design choices and alternative FatFont designs, their implementation and algorithms, a set of examples of their use, and a discussion on how and when FatFonts can be beneficial compared to existing alternatives.

2. RELATED WORK

Previous work from art and information visualization has made use of typography and related graphical techniques. In this section we discuss how these approaches are related to FatFonts. For clarity, other related work (e.g., information visualization techniques comparable to FatFonts) is cited in our discussion of example application scenarios.

2.1 Art, Typography, and Screening

Many artistic examples exist in which letters and digits have been used to create pictures. One well-known example is ASCII art, where a limited character set is creatively used to reproduce shapes and, taking the darkness of ASCII characters into account, greyscale images. In ASCII art textual and numeric characters are only the means to build an image; that is, single characters are not meant to convey meaning but to be experienced as a whole. In effect, the ASCII characters are used like pixels, not for the meaning they symbolize but to build up an image. With FatFonts our intention is to combine meaning in the individual characters with the capability of building images as a whole.

Artistic screening [21] is a development of the ASCII art idea. It allows halftoning through the use of any shape, including text and numbers. The primary purpose of artistic screening is quality halftoning where individual glyphs are beautiful in themselves. Intensity levels are predefined via a dot contour and, for a given shading, a chosen glyph modified by these predefined dot contours is utilized. In contrast to FatFonts, the shape alone is used for artistic purposes while the symbolic meaning is not relevant.

There are other artistic techniques that, similarly to Fat-Fonts, make use of text to convey meaning through images and words simultaneously, often in close relationship to each other. For instance, Xu and Kaplan developed an algorithm that shapes words according to their meaning: the letters of the word "cow" are distorted individually to form the shape of a cow [28]. Other works arrange text artistically into complex images to convey more content, even entire stories [16,

19]. This work and FatFonts share the dual representation of information (symbolic and visual) but FatFonts focus on the representation of quantitative data.

Other related artistic works include visual poetry [7] (e.g., Apollinaire's calligrams), and Islamic calligraphic art. More recently, *kinetic typography* videos (e.g., [20]) have become popular where typography is animated to highlight the content, tone, and attitudes of spoken dialogue or song lyrics. As with the previous examples, FatFonts differs from these in its goal of representing quantitative information.

2.2 Font Manipulation in Visualization

Early visualizations that required printing on paper through character-based dot-matrix printers or typewriters made use of fixed letters and numbers to create shapes useful to understand the data (e.g., Stem and Leaf plots [23] and related techniques [17]). We take these ideas further by modifying the glyphs themselves to create a meaningful visual overview. Text and numbers are also part of modern visualizations, but they are usually separate elements (e.g., labels), and their typeface is influenced by legibility or space efficiency, not the image. Examples of typography directly applied in information visualization are tag- and word-clouds [3, 11] and more recently Wordtree [26], that vary the font size and/or weight to represent information such as the word frequency in a text. These techniques convey relative quantities, rather than explicit numerical data. Finally, typographic maps make exclusive use of letters and typography to form city maps [2]. These maps are similar to FatFonts in that the outline and texture of the text both convey meaning, but the goal here is to relate city shapes and their names, not to communicate quantities as in FatFonts.

3. FATFONTS

FatFonts is a technique designed to bridge the gap between symbolic and visual representations through the manipulation of the shape of Hindu-Arabic numerals. In this section we describe the fundamentals, goals, and design decisions that constitute this novel technique.

3.1 FatFont Fundamentals

Glyphs as Symbols. Many of the advantages of modern numeral systems over unary systems stem from the abstract symbolic nature of the glyphs used. While unary systems enable a more literal and direct perception of a quantity (as shown in Fig. 1), Hindu-Arabic numerals can represent much larger numbers using less space and enable mathematical operations as well as comparisons. Importantly, modern numerals are less error prone because each numeral is visually distinct from its siblings. For example, the Hindu-Arabic numeral for seven has a significantly different shape than the numeral for eight, even though they are consecutive. Unary systems force us to count—a slow and error-prone process, especially with larger numbers [14]. This issue applies also to visualizations such as those in Fig. 1(b): the relative

333**33**₃ 444**44**₄

Figure 3: Numbers 3 and 4 in different typefaces.

amounts between the different quantities are easy to perceive but their exact values are more difficult to identify precisely. Another valuable feature of modern numeral glyphs is that their shapes can be modified significantly while remaining recognizable (see Fig. 3). FatFonts utilize this property by adjusting the area of numerals to reflect their numeric value.

Halftone. FatFonts and several of the artistic techniques described earlier rely on halftoning—providing "the impression of variable intensity levels by varying the respective surfaces of white and black within a small area" [21]. This means that the proportion of black and white within a region is aggregated by the human visual system and results in a certain level of intensity, even with irregular distributions and shapes of black marks within this local area. It is this effect that enables stippling techniques to provide the impression of continuous greyscale variation through exclusively black and white inking [15], and allows the perception of a larger "meta-picture" from many smaller ones in *photographic mosaics* [12]. Note that the definition of "small area" is flexible, since we can often perceive both the global image and its components within the same view. FatFonts parallels this by composing specially designed numerals to form "data mosaics" that can be perceived as images exploiting the ability to perceive both high- and low-level information from the same representation. The global image is the overview of all data and the individual numbers are the individual data.

3.2 FatFont Design Goals

Our initial FatFont design goals consider numeric, typographic, and visual aspects.

Familiarity. There are benefits to creating visuals that are easy to interpret with little or no instruction. Techniques that build upon previous knowledge are less likely to be misinterpreted, require less effort to learn, and may result in wider overall acceptance and use. Designing for familiarity generally implies the re-use of conventions and symbols, well-known to the general population.

Legibility. Legibility refers to the recognition of a single character without effort [22] and is a major concern in typography. Achieving legibility is fundamental for the design of FatFonts: if characters are difficult to identify or to distinguish from each other, this will negatively affect the ability to read the digits.

Readability. In typography, readability refers to the arrangement of characters in a body of text and the effort that is needed to read these characters in combination [22]. For prose it is crucial that larger units of characters such as words, lines, and paragraphs can be recognized. However, the readability of numbers differs from that of text. Tables with numbers are rarely read sequentially left to right but multiple jumps to any part of the table are part of how we read this kind of data. The readability of FatFonts depends on how digits are laid out to be perceived as belonging to the same or a different number.

Numeric Dynamic Range & Precision. The ability to represent a wide range of numeric data makes a visualization technique applicable to a variety of problems. Therefore, one design goal of FatFonts is to adapt to numeric data

types that vary in just a few steps (e.g., results from a Likert scale) and to data with larger ranges and smaller steps (e.g., percentages with decimals). Standard numeral systems have arbitrary precision, in contrast to visual representations that are often limited in the amount of variation that can be distinguished by the human perceptual system [24]. In Fat-Fonts, dynamic range and numeric precision are affected by decisions regarding the amount and position of digits with respect to each other to form a number.

Type Consistency & Visual Unity. Traditional typefaces can be characterized by a their visual unity. While each character within a typeface is unique, they all share some visual features which makes it possible to pick out numbers that belong to the same typeface in Fig. 3. Keeping a uniform look across characters is important to assure that every digit is recognized as part of the same whole and, simultaneously, contributes to a consistent and pleasant look. It is important for a visualization to be consistent in terms of how it is perceived as a whole—not just a collection of individual components.

Resolution. In data analysis it is often desirable to fit as much data as possible within the available presentation space so that a global view can be as detailed as possible. The overall resolution of a data visualization, however, is to a large extent determined by the limits of the display technology. Different FatFont variations can be designed to use different amounts of pixels per data point (or number) allowing a wide range of flexibility in the available resolution.

Spatial Precision. We define spatial precision as the accuracy with which a visual instance (e.g. a circle) can represent the location of a data point in two-dimensional space. For example, the area of a circle usually represents a data point associated with the circle's centre. However, when the area of the visual instance is large or shaped irregularly, it is difficult to exactly identify the location that the represented data refers to. With FatFonts, we aim for spatial precision to enable their use with spatially sensitive data sets.

Contrast. Visual dynamic range is another characteristic that affects the applicability of visualizations. It roughly corresponds to the maximum contrast that can be achieved in the representation of different colour or brightness levels and affects the perceptual properties of the visualization. For example, if the contrast is very low (i.e., if the highest representable level is not very different from the lowest), it will be harder to visually detect changes. In our Fatfonts variations we can achieve different levels of contrast.

3.3 Design Decisions

The goals above span a large design space with many alternatives. In the following sections we describe the principal design decisions that we made to generate four FatFont variants as a first exploration of this space.

Hindu-Arabic Numerals. Since we aim at familiarity, it is important to achieve high recognition value and ease of learning. In consequence, we decided to use Hindu-Arabic numerals (which are virtually universal) as the base for our four FatFont variants.

Multi-digit Numerals. The Hindu-Arabic system uses position to enable the representation of large numbers; numerals to the left indicate the multiplier for increasingly high orders of magnitude (in base 10). While this system would

fulfil our goals of achieving familiarity, and numeric precision, it has drawbacks regarding numeric range, readability, and spatial precision. A linear positional system grows always in one direction which can complicate the positioning of numbers in two-dimensional space: arranging data values of different magnitudes in uniform grids becomes difficult. Varying number lengths also negatively affects spatial precision because the centre location of long numbers is harder to determine. Maintaining a strong relation between one data point and the corresponding numeric value becomes difficult with high data densities (readability). Furthermore, the traditional positional system of Hindu-Arabic numerals supports the symbolic nature of numbers, but does not provide enough visual variation to create visual overviews that can be perceived in an instant, one of the main advantages of the visual system [5].

We therefore opted for a multi-digit system based on *containment* and *scaling*. Each digit is scaled corresponding to its order of magnitude, to maintain the relationship between the amount of ink and the represented number. For example, for the number 489, the area (including white space) that the 9 spans is one hundred times smaller than the area of the 4 and ten times smaller than the area of the 8 (see Fig. 4). Within their particular scale, the amount of ink of each digit directly corresponds to the numeric value it represents. Hence, this scaling preserves the linear relationship between the amount of ink and the represented number across multi-digit numbers.
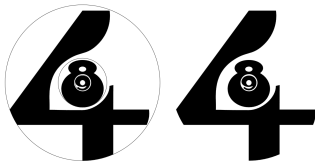
Figure 4: FatFonts representation of 489 (Miguta).

We decided to position digits that form a number based on recursive nesting. For instance, digit 8 within the 489 is nested within the space of the 4 and the 9 within the 8 (see Fig. 4). This recursive nesting of digits constrains FatFont numerals to a constant amount of space regardless of how many digits a number consists of, a property useful for visual layout. However, the system has limitations regarding dynamic range (the amount of digits defining one number). The digit for each order of magnitude is $\sqrt{10}$ smaller than the previous one, which makes it impractical to represent numbers beyond three digits; the smallest numbers turn into little more than a speckle. While this limits the dynamic range of FatFonts to 3 (or possibly 4) orders of magnitude, the nesting does make it possible to both symbolically and visually locate a data value at the appropriate position.

3.4   Four FatFont Variants

Four FatFont variants were designed with advice from typography experts and applied knowledge from typographic design authorities [9, 18]. The FatFonts typefaces were created in Adobe Illustrator to enable accurate measurements and adjustment of the area of digits and transformed into XAML scripts using Microsoft Expression Blend. This enabled us to computationally arrange and assemble FatFonts according to any data sets.

In our first FatFont type, MIGUTA, the digit glyph sits between two concentric circles, the inner circle of one tenth the area of the outer (see Fig. 4, left). Constraining the

ink to this area enables recursive nesting without occlusion between digits of different magnitudes, and it also places the "centre of gravity" of a number in the centre of its circular area, a beneficial property for spatial precision. Fig. 5 (top) demonstrates samples of Miguta use.

ROTUNDA is an alternative to Miguta with more flexibility regarding the positioning of multiple digits. Instead of forcing sub-digits to fit within the centre of their parent digit, in Rotunda they are positioned more flexibly according to their parent digit's white space (see Fig. 5). This sacrifices some spatial precision, but it enables the design of a FatFont type with cleaner lines that resembles traditional numeral fonts more closely, which may contribute to better legibility.

When arranging circle-based FatFont types in rectangular arrays, space is wasted between numbers. This results in reduced contrast and visual dynamic range because the negative space around the FatFont numbers appears in a colour (usually white) similar to low data values. The darkest halftone that can be generated with circular FatFont types such as Miguta or Rotunda is substantially less than pure black. Even a hexagonal packing of circular FatFont numbers showed only marginal improvement—not enough to justify the extra costs of processing and interpolating the images and data to conform to a hexagonal grid.

CUBICA, a square-based FatFont type (see Fig. 5) provides a better alternative. Cubica numerals use the same sub-digit allocation of Rotunda, but their square shapes allow more efficient packing. They only require a small inter-numeral gap for legibility which results in a higher contrast overall.

As a further exploration, we created GRACILIA, a FatFont type that is based on the sans-serif typeface "Helvetica" (see Fig. 5, bottom). Helvetica, created by the Swiss typographer Max Miedinger and Eduard Hoffmann in 1957 [13], is one of the most commonly used typefaces world-wide. In exchange for its familiar appearance, the Gracilia type has the lowest visual dynamic range and a non-square aspect ratio. Nevertheless, Gracilia may offer the best visual unity.

Flexible Weighting for Different Numeric Ranges. A direct mapping between the amount of ink and the represented value is not always desirable. For example, if the data being represented only covers the range between 36 and 65, using a linear mapping that starts in zero and ends in 99 will result in a low-contrast visualization. This is illustrated in Fig. 6, which shows a regular FatFont representation of a reduced numeric range version of the example data set shown in Fig. 5: instead of ranging from 3 to 95, it ranges from 36 to 65. To support a higher contrast, Gracilia allows *flexible weighting*, i.e., a customized mapping between weights and digits. Fig. 7 shows a use of a custom mapping of the Gracilia FatFont that enhances the visual contrast without any change in the symbolic values. Flexible weighting can also be used to establish arbitrary relationships between the visual and symbolic aspects of FatFont visualizations such as logarithmic or square-root mappings.

Considering "Zero" as a Special Digit. To preserve the linear relationships between the symbolic and visual representations we made a conscious decision to represent the digit 0 through blank space. However, this eliminates the distinction between actual blank space from data with the numerical value 0, and might make it difficult to determine how many 0 data points are located in a certain area. Currently our software offers the option to display an outline of
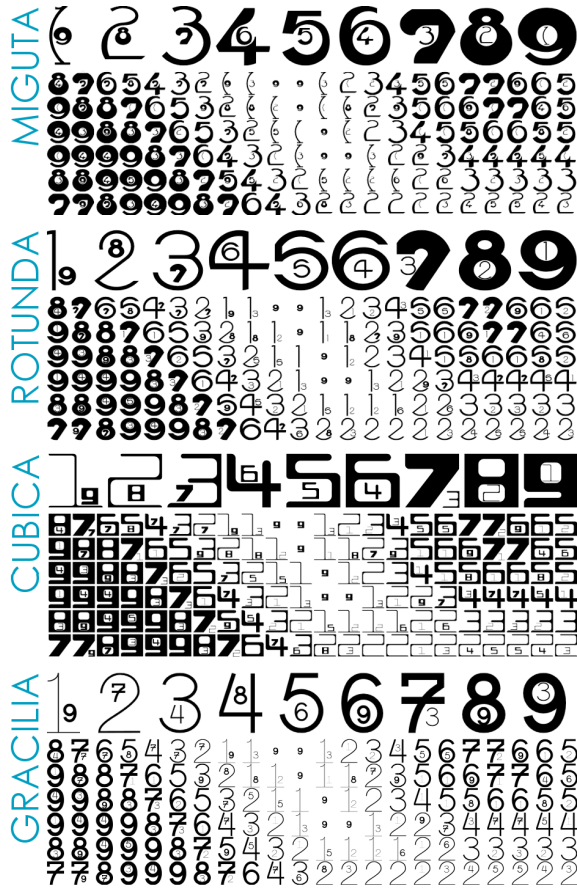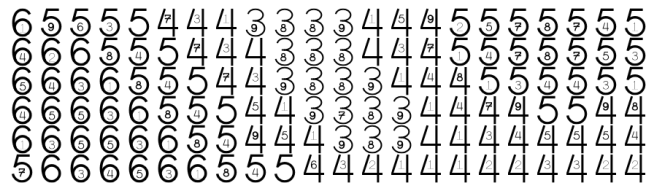
Figure 5: The four FatFont variants.

Figure 6: Representation of data with a limited range [36,65] in non-dynamic Gracilia FatFonts.

Figure 7: The same data as in Fig. 6 represented with dynamically weighted Gracilia FatFonts.

the space allocated for each data point (i.e., a circle as in Fig. 4) to indicate its presence. However, this can contribute to clutter or visual artifacts that may distort the overall perception of the visualization; other alternatives need to be explored in the future.

3.5 Design Challenges

Our FatFont typeface design iterations revealed trade-offs with regard to our design goals. Most importantly, there is a tension between achieving legibility and familiarity on the one hand (arguably best represented by Gracilia) and contrast and resolution on the other (arguably best represented by Cubica). While the consideration of spatial precision is important, we found that small variations in the centre of gravity of a number did not cause problems. Type consistency and aesthetics seem to depend more on designer skills than on balancing other goals, although overly constraining the design, as with Miguta, did affect our ability to make the digits familiar and recognizable.

Achieving visual unity proved to be another challenge. Since FatFonts rely on both the visual and symbolic perception of numeric glyphs, we aimed at avoiding interferences between the individual shape of digits with the overall perception of the visualization. At the same time, the legibility of individual digits had to be preserved. This was a challenge since the shape of some digits requires all the ink to be concentrated in a specific area which introduces artefacts within the overall visualization. The FatFont digit of 7, for instance, with its relatively high value and a simple narrow shape, requires the concentration of a large amount of ink (dark pixels), in a relatively constrained area. This problem

is most visible in the Rotunda type, where the digit 7 seems darker than 6, and as dark as 8. Although variations in the data still appear as gradual if seen from far, these effects should be avoided. We came close to resolving this problem with Gracilia using a middle stroke for the 7 that is common in Europe and Latin America.

3.6 Implementation and Technical Details

All FatFont glyphs are area shapes designed using Adobe Illustrator, which provides shape area measures through Telegraphic's Patharea Filter [1] plugin. The vector-based glyphs for each digit in each of the FatFont variants were then imported into XAML through Microsoft Expression Blend for use within a series of Microsoft WPF classes. Miguta, Rotunda, and Cubica sub-digits appear at fixed locations specified at design time, which are part of the font definitions and are therefore stored within the font classes.

To enable flexible weighting as described earlier, Gracilia glyphs are based on curves instead of filled shapes. The thickness of a curve's stroke is adjusted to achieve the desired amount of ink. Because curves for different digits cross, the required stroke thickness is non-linear and needs to be iteratively pre-calculated for each given weight-mapping. This is done once per mapping. The location of sub-digits is also calculated algorithmically to avoid overlap, since variation in stroke thickness affects the optimal location of sub-digits.

4. VISUALIZING DATA USING FATFONTS

In the following section we describe three examples of how FatFont typefaces can be applied to problems typically addressed by visualization techniques, especially when an overview as well as numerical detail of large data sets is required. Through this discussion of initial FatFonts example scenarios we hope to encourage experts to envision other potential applications where FatFonts can be beneficial.

4.1 Scalar Fields

We refer to scalar fields as data defined by a spatial position on a plane and an attribute assigned to this position which is typically mapped to a visual variable such as colour, shape, size, or, in our case, FatFonts. Figures 8 and 9 show examples of scalar fields represented through FatFonts. Figure 8 represents the maximum wave amplitudes (in cm) for the devastating March 11, 2011 Japan

---

[1]http://www.telegraphics.com.au/sw/

tsunami. We chose Gracilia in this example for its visual unity. Figure 9 shows Sicily's topographic elevations normalized to their highest elevation (the highest point on Sicily, Mount Etna 3320m $\hat{=}$ 99) in Cubica for higher contrast.

These examples illustrate how FatFont scalar field visualizations support many of the basic analytic tasks in visualization [1]. We can identify local maxima represented by the darker parts of the visualization (visual aspect) and compare the highest numeric values (symbolic aspect). Different local maxima can be directly compared to determine their order and distribution (Mount Etna is 68% higher than the Pizzo Carbonara in the Madonie range). We can even find low elevation routes to cross Sicily from one end to another.
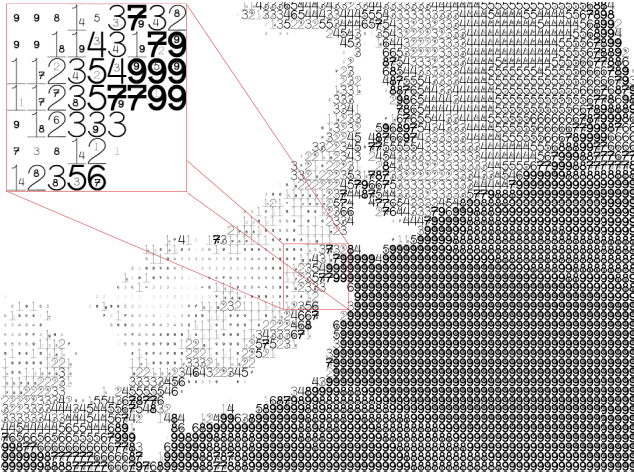

Figure 8: Maximum wave amplitudes for the Japan 2011 tsunami. Amplitudes were clipped at 99cm. Data adapted from NOAA; http://www.noaa.gov/.
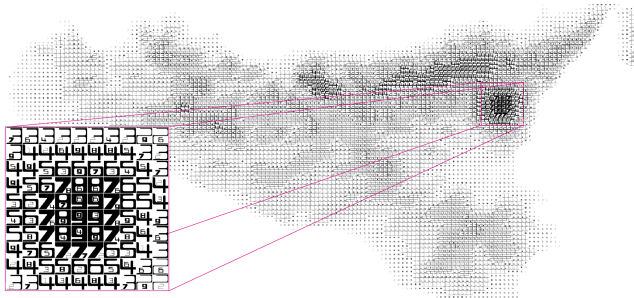

Figure 9: Sicily's topographic elevation normalized by its maximum (3320m $\hat{=}$ 99). Data adapted from NASA; http://visibleearth.nasa.gov/.

FatFonts are particularly suitable to high-resolution display technology (see Fig. 10 and Video Fig. A) where they enable a high-level overview of the data (looking at the display from a distance) while providing a detailed view of data values on demand (moving closer to the screen). Even without such technology, interaction techniques such as magnification lenses or pan-and-zoom can be provided to support overview-plus-detail views (see Fig. 11).

4.2 Highlighting Numerical Changes

For some analysis tasks it can be important to highlight dynamic changes in data. Large amounts of dynamically changing data represented with FatFonts can be overwhelming; this is true for most visualizations, particularly if they are glyph based. However, for highlighting slow and subtle
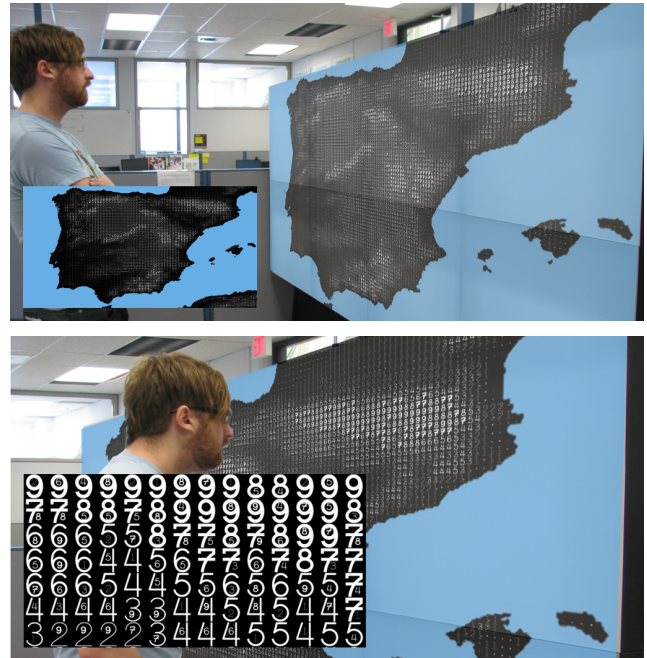

Figure 10: FatFonts on a high-res wall.

changes in data, FatFonts can be beneficial since Hindu-Arabic numerals differ significantly in shape and even small changes in a large data set become salient (see Video Fig. B and C). While an analyst might miss subtle colour changes, small changes in FatFont numbers are quite conspicuous, even if they occur in lower orders of magnitude. This might prove valuable in monitoring scenarios such as spatially distributed sensors for tracking small (but important) changes; e.g., air composition in different areas of a chemical facility.

4.3 Combination with Existing Techniques

FatFonts can be combined with other visualization techniques which can enhance existing techniques and help address some of the limitations of FatFonts mentioned above. For instance, detail-and-context techniques such as lenses [10] can complement FatFonts to support high resolution and dynamic range and numeric precision, while enabling the comparison of numeric data values (see Fig. 11 and Video Fig. D). Similarly, FatFonts can be combined with existing visualization techniques such as grey- or colour-scales in different visualization layers interactively accessible via Magic Lenses [6] (see Fig. 12).
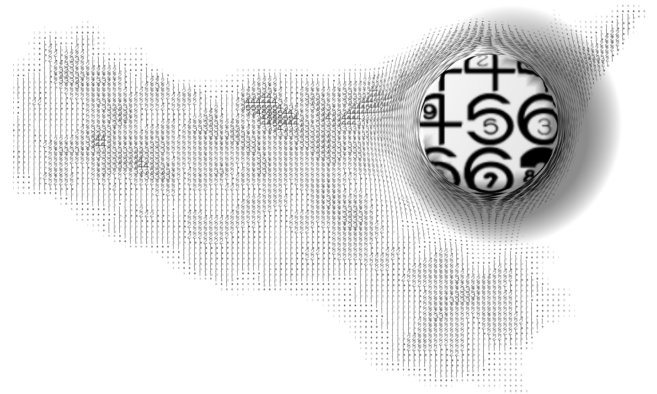

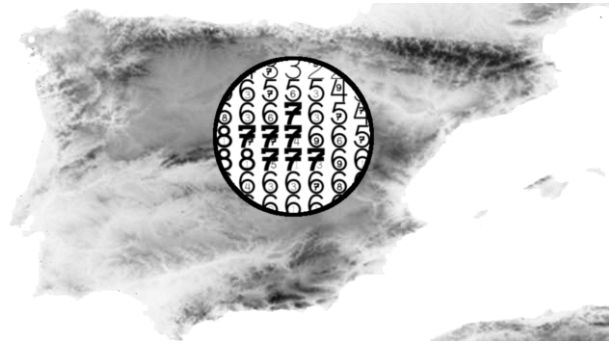**Figure 11: Using lenses on a FatFont visualization.**

Figure 12: Combining greyscale visualizations with FatFonts (mock-up).

5. DISCUSSION

The advantages of FatFonts stem from their hybrid nature. While alternative approaches such as colour-scale maps or numeric tables will always have their place, FatFonts have the ability to *simultaneously* address several of their limitations. For example, for generating an overview visualization that shows the general distribution of a continuous variable, a colour- or greyscale representation would probably be the best choice. However, colour- and greyscales are limited in the number of data levels that can be reliably distinguished [24, 25]. In this particular aspect, the 99 levels of two-digit FatFonts outperform the ability of the human visual system to distinguish differences in a standard colour map. Naturally, a table with numbers can easily provide more than two digits of precision, yet this representation does not provide the advantages of a visual overview of the data (e.g., being able to rapidly direct attention to the highest-valued areas). In the following we discuss the advantages and limitations of the FatFonts technique.

FatFonts: Fully Labelled Visualizations. A FatFont representation of scalar fields is akin to a visualization where each value is labelled; the raw data points themselves become visual variables. For large data sets that require an accurate identification of exact values, this is can be an advantage. The alternative of providing dynamic labels interactively does not allow for all data values to be present at the same time due to occlusion and overlap problems. Furthermore, analyzing the details by systematically hovering over specific data regions can be cumbersome and generates a considerable memory load. FatFonts do not require this type of interaction, freeing this option for other tasks (e.g., annotation). In addition, a FatFont visualization retains its properties when printed in static media.

FatFonts vs. Legends. Common representations of scalar fields are colour- or grey scales that are linked to values through legends (e.g. [8]). FatFonts solve some of the well-known problems of such visualizations. There are only a limited number of colours that can be accurately distinguished; far fewer than the 99 reliably distinguishable levels that two-level FatFonts provide. Furthermore, it is difficult to provide scales with many colours that have a perceptual ordering [25], and our perception of colour and brightness is affected by the local context. This makes it difficult to compare two different colour levels that are spatially separated in a visualization. The FatFonts technique allows numeric data to be perceived visually and symbolically in an accurate and less error-prone way. It also enables proportional comparisons between data, which are practically impossible in a colour map (which colour is double that of another colour?). Generally, FatFont representations do not require colour, and are, thus, less affected by colour blindness.

Resolution. FatFonts visualizations require multiple pixels per data point. This reduces the amount of data that can be displayed compared to regular greyscale maps that require a minimum of one pixel per data point. Furthermore, there is a strong trade-off between the resolution and the desired numerical dynamic range of FatFonts; the more digits per data point represented, the more pixels each number will require. Nevertheless, FatFonts degrade gracefully with downscaling toward a visualization comparable to a greyscale representation if a decent interpolation is provided. Furthermore, when reducing the space allocated to each FatFont number, it is the least important information, the least significant digit, that becomes illegible first.

Moreover, resolution is, in many cases, not the only criteria for choosing a visualization technique. For example, in a scenario such as the high-resolution wall shown in Fig. 10, resolution is not an issue when the viewer is at a medium distance from the display, where the digits are not distinguishable but she can perceive the equivalent of a grayscale image. If the symbolic nature of the data is required, the viewer can come closer (as a natural detail-in-context technique). In a similar fashion, FatFonts can be combined with a zoomable interface (e.g., the infinite canvas from Pad++ [4]) to provide a smooth transition between image- and symbol-supported tasks.

Hybrid vs. Mixed Techniques. Excel's *conditional formatting* feature (which can be considered a precursor of FatFonts), can turn a table with numeric values into an image that combines both numbers and grey- or colour scales (see Fig.13). However, the numbers do not provide the grey scale themselves but it is applied to the background of table cells. While this provides an overview combining both numeric values and their corresponding greyscale, there are immediate foreground/background issues. First, choosing the shade of the text is not trivial: full greyscale contrast will render some numbers invisible. Numbers could be coloured individually according to the background colour, but this would interfere with the perceived colour itself. Second, the amount of ink of regular numeral typefaces is not linear with respect to their digit values, which adds noise to the image. Third, in FatFonts the glyph itself is the label, which allows the top-level digit to be larger than its Excel counterpart, which has to be embedded within a square. It should also be noted that higher precision for conditional formatting means adding more digits within the cells which, in turn, forces the aspect ratio of the table to change, and, with it, the visual overview. While Excel's conditional formatting does not yet offer the symbolic and visual integration of numeric data values into single glyphs, its existence as a commercial application suggests a need for techniques such

Figure 13: MS Excel's conditional formatting.

as FatFonts that offer symbolic data details integrated into a general overview visualization.

Occlusion. Because digits rely on shape to be recognizable, FatFonts are sensitive to visual interference by occlusion. This means that superimposing other elements of a visualization on top of FatFont numbers is likely to interfere with the symbolic reading of data. Some methods to reduce such problems include lowering the opacity, or applying glow or shadows. In general, however, we recommend avoiding overlap if possible.

FatFonts integrate symbolic and visual information, and it is in this hybrid space that they create opportunities. The presented examples show a range of possible applications of FatFonts—from well-known problems associated to the perception of greyscale levels, to an incursion into dynamic data visualizations. The exploration of these scenarios has led us to a better understanding of the potential of the technique, but also of its limitations. In general we can conclude that FatFonts are useful when: (1) absolute values need to be understood in context, (2) small differences or changes in data are important, (3) perceptual biases can induce to error, and (4) spatial resolution of the data is not the primary concern. At the same time, the iterative design process of designing a variety of FatFont typefaces has been valuable for understanding how the different design goals that we identified interact with each other, and how typeface design can address some of the trade-offs.

6. CONCLUSION

We have introduced the idea of FatFonts, a novel visualization technique that integrates the symbolic and visual aspects of numbers. As a main contribution we provide a first exploration of the design space of this technique. As part of this effort we have presented four FatFont typefaces that represent different variations of the FatFonts technique. We provided a nesting technique that enables the representation of multi-digit FatFont numbers in a compact and consistent way and discussed a set of examples from different problem domains. As a secondary contribution we have presented a set of design goals that consider the numeric, typographic and visual aspects of the FatFont technique and that can guide the use of numeric typefaces in the context of information visualization.

The FatFonts technique is particularly promising for large data sets where visual overview as well as numeric data details are important. FatFonts are suitable for display on large high-resolution screens, where moving forward or backward can provide a fluid way to switch focus from the overview to the details of the data as needed.

Our exploration of numeric typeface manipulation for visualization purposes is by no means exhaustive. Other application scenarios for FatFonts need to be explored, as well as alternative nesting arrangements and non-linear relationships between the number and the amount of ink. On a practical level, we are working to provide an easy-to-use programming interface that enables the use of FatFonts in existing visualization systems and the web.

7. REFERENCES

[1] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proc. of InfoVis*, pages 111–117, 2005.

[2] Axis Maps. Typographic maps. http://www.axismaps.com/typographic.php, 2010. Retr. 07-2011.

[3] S. Bateman, C. Gutwin, and M. Nacenta. Seeing things in the clouds: the effect of visual features on tag cloud selections. In *Proc. of Hypertext and hypermedia*, pages 193–202, 2008.

[4] B. B. Bederson and J. D. Hollan. Pad++: a zooming graphical interface for exploring alternate interface physics. In *Proc. of UIST*, UIST '94, pages 17–26, 1994.

[5] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps.* The University of Wisconsin Press, 1983.

[6] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses. the see-through interface. In *Proc. of SIGGRAPH*, pages 73–80, 1993.

[7] W. Bohn. *The aesthetics of visual poetry, 1914-1928.* University Of Chicago Press, 1993.

[8] C. A. Brewer. Proc. of the section on statistical graphics. pages 55–60. American Statistical Association, 1999.

[9] R. Bringhurst. *The Elements of Typographic Style*, chapter 2: Rhythm & Proportion, page 25. H&M Publishers, 2005.

[10] M. Carpendale and C. Montagnese. A framework for unifying presentation space. In *Proc. of UIST*, pages 61–70, 2001.

[11] J. Feinberg. Wordle. http://www.wordle.net/, 2009. Retrieved Sept. 2011.

[12] J. Francis. The history of photo mosaics. http://www.digitalartform.com/archives/2004/12/history_of_phot.html, 2004. Retr. 09-2011.

[13] G. Hustwit. Helvetica. Documentary Film 'Helvetica': http://www.helveticafilm.com/, 2007. Comment by Wim Crouwel about the Helvetica typeface.

[14] G. Ifrah. *The Universal History of Numbers: From Prehistory to the Invention of the Computer.* Wiley, 2000.

[15] S. Kim, R. Maciejewski, T. Isenberg, W. M. Andrews, W. Chen, M. C. Sousa, and D. S. Ebert. Stippling by example. In *Proc. of NPAR*, 2009.

[16] T. Kirton. Artistic canvases for gestural and non-linear typography. In *Proc. of Computational Aesthetics*, 2011.

[17] J. B. Kruskal and J. M. Landwehr. Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, 37(2):162–168, May 1983.

[18] E. Lupton. *Thinking with Type: A Critical Guide for Designers, Writers, Editors, & Students.* Princeton Architectural Press, 2010.

[19] R. Maharik, M. Bessmeltsev, A. Sheffer, A. Shamir, and N. Carr. Digital micrography. *Trans. on Graphics*, 2011.

[20] J. Moody. Say what again. http://www.jarrattmoody.com/, 2007. Retr. 09-2011. Contains violence and coarse language.

[21] V. Ostromoukhov and R. D. Hersch. Artistic screening. In *Proc. of SIGGRAPH*, pages 219–227, 1995.

[22] A. Poole. Literature review: Which are more legible: Serif or sans serif typefaces? http://www.alexpoole.info/academic/literaturereview.html, April 2005. Retr. 09-2011.

[23] J. W. Tukey. *Exploratory Data Analysis.* Addison Wesley, 1977.

[24] C. Ware. Color sequences for univariate maps: Theory, experiments and principles. *IEEE Computer Graphics and Applications*, 8(5):41–49, 1988.

[25] C. Ware. *Information Visualization—Perception for Design*, chapter 3: Lightness, Brigntness , contrast, and Constancy, pages 73–102. Morgan Kaufmann, 2004.

[26] M. Wattenberg and F. B. Viégas. The word tree, an interactive visual concordance. *IEEE TVCG*, 14:1221–1228, November 2008.

[27] Wikipedia contributors. Numeral system. Wikipedia: http://en.wikipedia.org/w/index.php?title=Numeral_system&oldid=351943580, 2010. Retr.09-2011.

[28] J. Xu and C. S. Kaplan. Calligraphic packing. In *Proc. of Graphics Interface*, pages 43–50, 2007.