Feature Article

Glyph Visualization: A Fail-Safe Design Scheme Based on Quasi-Hamming Distances

Philip A. Legg ▪ *University of the West of England*

Eamonn Maguire ▪ *CERN European Laboratory for Particle Physics*

Simon Walton and Min Chen ▪ *University of Oxford*

In many spatial and temporal visualization applications, glyphs provide an effective means for encoding multivariate data. Glyph-based visualizations are ubiquitous in modern life because they make excellent use of the human ability to learn abstract and metaphoric representations, facilitating instantaneous recognition and understanding. However, because glyphs are typically small, they are vulnerable to various perceptual errors. Glyphs are often designed with a high degree of similarity to facilitate mapping consistency, semantic interpretation, learning, and memorization. As Figure 1 shows, as the size of the graphs decrease, the glyphs become indistinguishable. Similarly, color defects from poor printing or uncalibrated display screens could make it difficult to differentiate between glyphs.

Although many glyph designers attempt to account for perception and legibility, they may also wish to incorporate their own creativity and intuition as an artist. Thus, a delicate balance exists between the science and the art of effective glyph design. This poses some challenging research questions for glyph visualization: Is there a theoretical framework to encompass various design guidelines? Is there a systematic approach to designing a fail-safe glyph set? This work is a direct attempt to answer the second question, but we also connect glyph-based visualization with information theory, which has been considered a candidate framework for visualization in previous work.[1]

In this article, we explore the concept of Hamming distance, a well-established measure from information theory that underpins the study of codes to support error detection and error correction by the receiver, without the need for corroboration from the sender. Specifically, we introduce the concept of a quasi-Hamming distance in the context of glyph design. We examine the feasibility of estimating the quasi-Hamming distance between a pair of glyphs and the minimal Hamming distance for a glyph set. This measurement enables glyph designers to determine the differentiability between glyphs, facilitating design optimization by maximizing distances between glyphs under various design constraints. We demonstrate the design concept by developing an event visualization tool that can depict the activities of multiple users of a file system. Our evaluation shows that the concept of quasi-Hamming distance allows us to design glyphs that significantly reduce the vulnerability of glyph-based visualization. We hope

To address the various perceptual errors common to glyph-based visualizations, this article introduces the concept of a quasi-Hamming distance in the context of glyph design and examines the feasibility of estimating the quasi-Hamming distance between a pair of glyphs and the minimal Hamming distance for a glyph set.

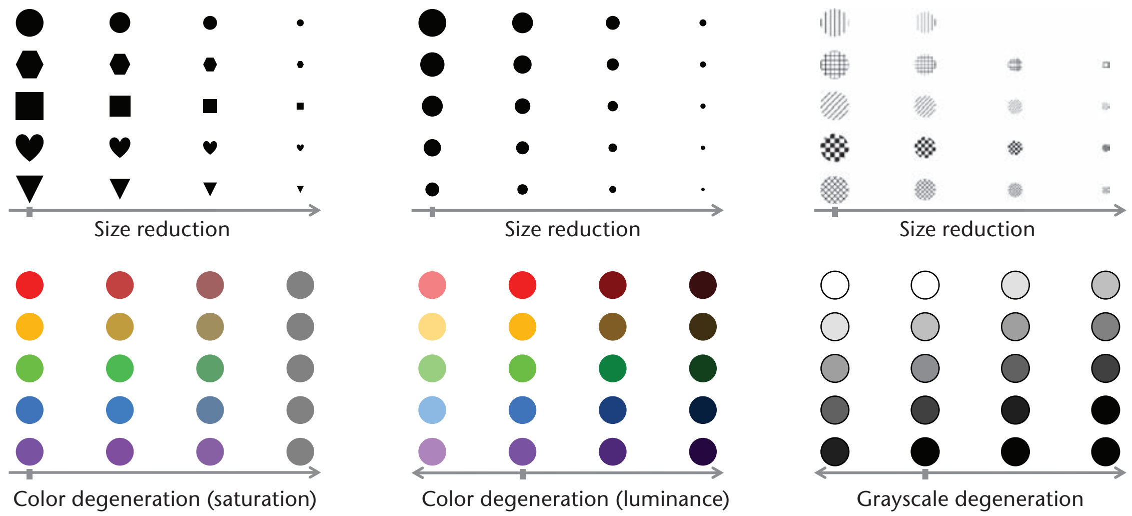

Figure 1. Different types of quality degeneration applied to glyphs, each of which is encoded using a single visual channel. When their size, saturation, or luminance change, glyphs can become more difficult to differentiate. The original image quality is indicated by the rectangular marker on the *x* axis.

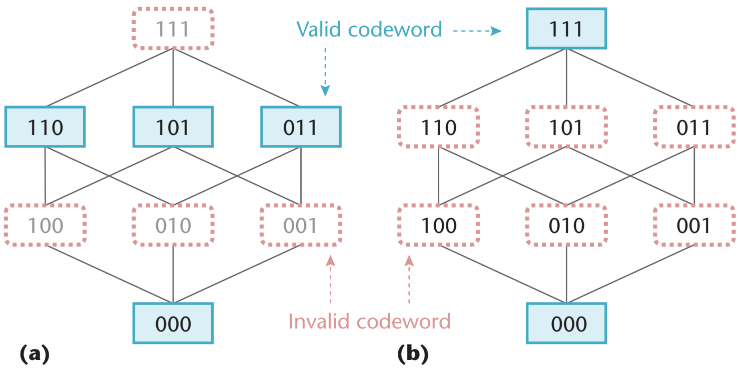

Figure 2. Two 3-bit codes. (a) The first code can detect 1-bit errors, and (b) the second code can detect 2-bit errors and correct 1-bit errors.

that this new concept will encourage designers to consider systematically the need to empower visualization users to detect and correct potential perception errors.

Hamming Distance

In information theory and data communication, a code consists of a finite set of *codewords*, each of which is a digital representation of a letter in an alphabet. In the context of binary encoding, *Hamming distance*, proposed by Richard Hamming in 1950, is a measure of the number of bit positions in which two codewords differ.[2] Considering all pairs of codewords in a code, the minimal distance is referred to as the code's minimal Hamming distance. (In the literature, the word minimal is often confusingly omitted). In communication, there are two main strategies for handling errors that occur during transmission:

- Automated error detection lets the receiver discover if an error has occurred and request a retransmission accordingly.
- Automated error correction enables the receiver to detect an error and deduce what the intended transmission must have been.

Hamming defined the following principle:

> A code of $d + 1$ minimal Hamming distance can be used to detect d bits of errors during transmission. A code of $2d + 1$ minimal Hamming distance can be used to correct d bits of errors during transmission.[2]

For example, given a 3-bit code, as illustrated in Figure 2, there are eight possible codewords. We can select a subset of these codewords to construct a code with its minimal Hamming distance equal to 2 or 3 bits. Figure 2a shows one such code that has four code words and is of 2 bits Hamming distance. This code can detect 1-bit errors because any change in a valid codeword by 1 bit would result in an invalid codeword, which would lead the receiver to discover the error. Figure 2b shows another code with two codewords and of 3 bits Hamming distance. It can detect 2-bit errors and correct 1-bit errors. When a valid codeword (such as 111) is changed by 1 bit during transmission (such as 110), the receiver can detect such an error and recover the intended codeword based on the nearest-neighbor principle. If a 2-bit error occurred during transmission, the receiver would be able to detect the error but could not correct it. Nevertheless, if 2-bit errors are likely to occur, then this should either be used as only an error-detection code or a code with a longer Hamming distance should be used instead.

Quasi-Hamming Distance for Glyph Design

A set of glyphs is a code, and each glyph in the set is a valid codeword. During visualization, there can be errors in displaying or perceiving a glyph. If a viewer can detect that a perceived glyph is not correct, a conscious or unconscious effort can be made to fix the error. Conscious effort, which is an analogy of error detection and repeated transmission, may include zooming in for a closer look or consulting a legend. Unconscious effort, which is an analogy of error correction, can be the result of Gestalt effects (the nearest neighbor)[3] or a combined judgment involving multiple visual components (redundancy).[4]

Figure 3 shows two example glyph sets, each with eight codewords. Given the two display errors depicted on the left—an arrow glyph is skewed in Figure 3a and a shape glyph is occluded by another shape in Figure 3b—we can detect both errors easily. The error with the arrow glyph may need some conscious effort, whereas the shape error can often be corrected unconsciously. This suggests that it is possible to establish a conceptual framework, similar to Hamming distance, for error detection and correction in glyph-based visualization.

However, measuring the distances and errors in visual perception is clearly not as simple as measuring those represented by binary codewords. We thereby propose an approximated conceptual framework based on the principle of Hamming distance and call it *quasi-Hamming distance* (QHD). The term quasi implies that the distance measure is approximated, as is the quantitative measure of perceptual error.

QHD can be considered a kind of measurement of "perceptual distance" between two glyphs. QHD offers several benefits:

- It connects glyph-based visualization to information theory through an important and widely used concept (Hamming distance) in computing and communication.
- It relates the need for accurate perception of glyphs to error detection and correction by the receiver.
- It facilitates quantitative measures semantically equivalent to Hamming distance and its mathematical implication.

The main research questions are therefore as follow: Can we establish a measurement unit common to both measures? How do we obtain such measurements?

For the first question, we can utilize a bit as the

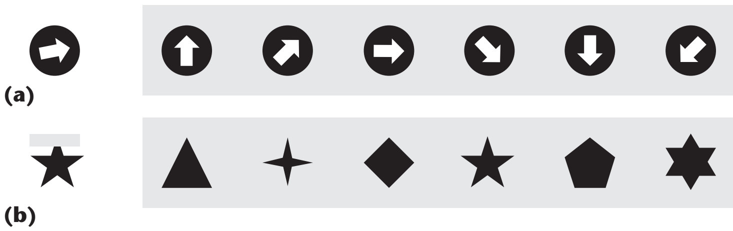

Figure 3. Examples of error detection and correction in glyph-based visualization. (a) In the first example, the orientation of the glyph on the left is distorted. A viewer may sense this and consult the legend for correction. (b) In the second example, a viewer may unconsciously perceive the glyph on the left as a star shape due to Gestalt effects and a priori knowledge about the glyph set, even though the top point of the star is missing because of occlusion.

common unit for both distance and error measurement. Let us first consider an ordered visual channel, such as brightness or length, as a code C. Theoretically, C can have a set of codewords $c_1, c_2, \ldots, c_n$ such that the difference between two consecutive codewords is the just-noticeable difference (JND) of this visual channel.

We can define the QHD between each pair of code words c_i and c_j as $|i - j|$ bits. If c_i is mistaken for c_j, we can call this a d-bit error, where $d = |i - j|$. Now let us extend this concept to a less ordered visual channel (hue) or an integrated channel (color). Theoretically, we can construct a code C by uniformly sampling the visual channel's space (for example, the CIE Lab color space) while ensuring that every pair of samples differ by at least the JND of this channel. These codewords, or samples, can be organized into a network, where the distance between any two codewords can be approximated proportionally according to the JND (JND = 1 bit). Note that the possible perception error rate with a code that maximizes the number of codewords based on JND is likely to be high. In practice, a glyph set is designed based only on a small subset of samples in a visual channel or more commonly in the multivariate space of several visual channels. Hence, a QHD measure based on JND would be too fine to use in practice, although in a longer term, JND can provide an absolute reference measure once we have obtained such measures for most visual channels in visualization.

Measuring QHD

This leads us to the second research question: Given a glyph set, how can we measure the distance between glyphs? We can consider using the following methods:

1. *Estimation by expert designers.* This practice has always existed in designing exercises, such as

for traffic signs and icons in user interfaces. To formalize this practice, designers can explicitly estimate and label the distance between each pair of glyphs in a glyph set. Although this approach may be most convenient to the designers, its effectiveness depends on the experience of the designers involved, and it is rather easy to overlook certain types of display and perception errors.

2. *Crush tests*. Introduced by Eamonn Maguire and his colleagues,[5] crush tests rescale glyphs to lower pixel resolutions to assess the preserved detail. We can simulate different causes of errors, such as those illustrated in Figure 1, and determine at which level of degeneration glyphs may become indistinguishable. The corresponding level of degeneration can be used to define the QHD. Although this approach yields a more consistent QHD estimation, more research would be required to compile a list of different causes of errors and define coherent levels of degeneration.

Quasi-Hamming distance can be considered a measurement of "perceptual distance" between two glyphs.

3. *Task-based evaluation.* Similar to method 2, we can simulate different visualization conditions, enlist users to perform their tasks, measure user performance, and transform performance measures to QHD. This approach is perhaps most semantically meaningful for a particular glyph set in a specific application context. However, the performance measures collected may exhibit many confounding effects, and the specifics of the application may mean that only a small number of users are available for evaluation.
4. *User-centric estimation*. We may conduct a survey among human participants about how easy or difficult it is to differentiate different glyphs. By removing the task dependency necessary for method 3, more participants can be involved in such a survey, yielding a more reliable QHD estimation.
5. *Computer-based similarity measures.* A variety of image similarity measures already exist in the literature.[6] In the long term, it is likely that we will be able to find measures that are statistically close to user-centric estimation, although there is not yet a conclusive confirmation about optimal image similarity measures, and there are hardly any metrics specially designed for measuring glyph similarities.

To demonstrate the feasibility of estimating QHD, we conducted two proof-of-concept experiments based on methods 4 and 5. We considered a task-based evaluation (method 3) as a possible approach because the glyph set under consideration is a file-activity visualization. However, this additional cognitive load may distort the glyph similarity assessment. Instead, for the purposes of this study, method 4 allowed a much more general audience to participate in the study, and method 5 let us corroborate the two methods.

We conducted a survey with 20 participants, all of whom are either employees or students at the University of Oxford. About half the participants had encountered glyph-based visualization previously. The results of one participant were considered an outlier and were not included in the statistics. After a brief introduction by one of the coauthors of this article, we asked the participants to rate how well they could differentiate 104 pairs of glyphs, on an integer scale between 0 and 10.

The 104 stimuli pairs were divided into three main categories: eight reference pairs, 48 primitive pairs, and 48 application-specific pairs. The eight reference pairs were designed to define the minimal and maximal QHD in the context of this work. In four reference pairs, the two glyphs are extremely difficult to differentiate (minimal distance). In the other four pairs, the two glyphs can be differentiated with undisputable ease (maximal distance).

We divided the primitive pairs into eight groups: hue, shape, components, connection lines, luminance, size, texture, and orientation. In each group, different pairs feature graphs with different perceptual distances, which let us obtain the human-centric estimation of the QHD for basic visual channels individually. The application pairs contain glyphs designed for our application case study, which we will discuss in detail in the "Case Study" section.

The 96 primitive and application-specific pairs were mixed together in a randomized order. The eight reference pairs were placed at positions 1, 2, 35, 36, 69, 70, 103, and 104 to help the participants regularize their scores and to enable us to check temporal consistency. See the online supplementary materials for further details about the questionnaire and stimuli grouping (https://extras.computer.org/extra/mcg2017020031s1.pdf) and for the full

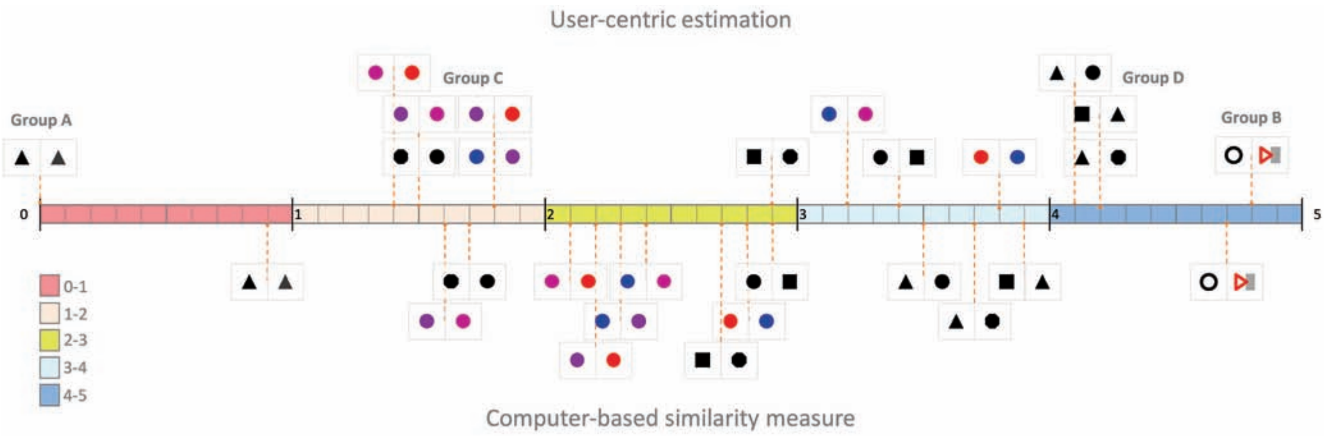

Figure 4. Sample of results for the primitive glyph pairs used in both the user-centric estimation (top) and by computer-based similarity measure (bottom). Low values indicate pairs that are difficult to differentiate, and high values indicate easy to differentiate. Examples from groups A–D are highlighted, where group A is hard to differentiate, group B is easy to differentiate, group C is differentiated by color, and group D is differentiated by shape. The color legend shows the QHD ranges, where red indicates a low QHD ($0 \leq$ QHD ≤ 1) and blue indicates a high QHD ($4 \leq$ QHD ≤ 5).

survey results (https://extras.computer.org/extra/mcg2017020031s2.xlsx).

As we mentioned earlier, we divided the category of reference pairs into two groups. Group A consists of four pairs of very similar glyphs, and group B consists of four pairs of very different glyphs. We expected that participants would assign low scores (difficult to differentiate) to those in group A and high scores (easy to differentiate) to those in group B, respectively. We placed one pair from group A and one from group B at regular intervals. The average scores for the four pairs in group A were 0.0, 0.4, 1.4, and 2.8, respectively. Those for group B were 9.3, 9.0, 8.9, and 9.7, respectively. These results indicate that they statistically served as references for the minimal and maximum QHD in this survey.

The category of primitive pairs consists of eight groups (C–J) for estimating QHD in relation to the eight visual channels (C: hue, D: shape, E: components, F: connection lines, G: luminance, H: size, I: texture, and J: orientation). Figure 4 shows a sample of the groups. (Full group details are available in the supplementary material.) Each group has six pairs of stimuli, facilitating a pairwise comparison of four different codewords for each channel. For the hue and luminance channels, after choosing the first and fourth codewords, we used a perceptually uniform color model (Hunter's lab) to determine the second and third codewords at 50 and 75 percent distance from the first, respectively. The upper part of Figure 4 shows a small selection of the survey results, where we converted the [0, 10] score range to a [0, 5] QHD range. We consider a QHD of less than 2 as potentially risky for error detection, and a QHD of less than 3 as potentially risky for error correction.

Table 1. Average QHD results for the primitive glyph pairs by group.*

Group	Average QHD	
	Human	Computer
E (Component)	3.5	3.4
D (Shape)	3.4	3.5
J (Orientation)	3.0	1.6
F (Connection)	2.8	2.4
I (Texture)	2.2	3.3
C (Color)	2.2	1.5
H (Size)	2.1	4.2
G (Luminance)	1.2	2.4

**These results were obtained from human participants and from a computer-based algorithmic similarity measure. They are ordered according to the human estimation.*

In our second experiment, we measured the similarity between each pair of glyphs using a computer-based metric. To calculate this, we developed a metric based on weighted invariant image moments,[6,7] a well-established approach in computer vision that is widely used for image similarity. The metric incorporates both pixel color and spatial occupancy to assess the difference between two images. The former captures a variety of feature differences (such as color, luminance, size, and orientation) and is defined as the mean Euclidean distance between all corresponding pixels in the two images representing the pair of glyphs. The latter captures location-invariant features (such as spatial occupancy) and is defined as the difference between the numbers of pixels with ≤ 80 percent luminance. Both difference measures are first normalized to the [0, 1] range and are then scaled to the same QHD range as the survey (with the same min, mean, and max) before being combined into a single metric. The lower part of Figure 4 shows the computed similarity measures for the same selection of stimuli pairs.

Table 1 shows the average QHD results for both the human-centric study and the computer similarity measure by group. With the human participants, shape, component, and orientation all scored a QHD ≥ 3, which suggests they are well separable visual channels. Luminance, however, was the only group to score below QHD < 2, suggesting it is not well separable. While some visual channels were also scored similarly by the computer-based similarity measure (shape, component, and connection), other channels differed from the user feedback (size and orientation). This shows that further research needs to be conducted in this area regarding how a computer can understand human visual perception.

Case Study: Visualizing File-System Events

The problem of visualizing file systems plays a significant role in the short history of computer-assisted visualization. In 1991, motivated by the need to visualize a file system's structure, Brian Johnson and Ben Shneiderman published their seminal paper on tree maps.[8]

Today, file systems are much larger and contain many more files; they are also shared by many more users and have many more events. One important purpose of a file system is to support collaborative activities, such as sharing files within multipartner projects and developing software among a team of programmers. Although there are text-based mechanisms for recording events in relation to a file system or a specific folder, the amount of data contained in typical log files can easily escalate to the point where it becomes too overwhelming for anyone to read on a regular basis. To the best of our knowledge, there are no effective visualization techniques that allows users of such collaborative environments to observe events effectively.

In this case study, we designed and developed a novel glyph-based visualization tool for observing events in a file system. There were several technical challenges. First, the file system's hierarchical nature needs to be depicted so that the spatial context of where a particular event has occurred can be identified. Second, the temporal information about events needs to be conveyed so that the activity ordering can also be observed and reasoned. Third, there are a wide range of activities (such as copying and modifying a file) that are typically performed that would need to be distinguishable in a visualization. Finally, the visualization should support collaborative environments by depicting activities from different users.

Designing File-System Event Glyphs

Like the design of most visual representations, glyph design needs to achieve a balance among many factors relating to the data, user, task, and application. For this application, we consider that the volume of data is high, although it is expected to be filtered in some way, such as for a specific portion of the file system, specific file types, or specific user groups. Nevertheless, the glyphs are expected to be relatively small and plentiful on a display screen. The visualization tasks are primarily routine observation and external memorization of the events in a file system. The users are expected to be regular users who will have the motivation, ability, and time to familiarize themselves with multivariate glyphs, although any metaphoric encoding will benefit learning and memorizing glyphs.

We considered the QHD concept throughout the design, development, evaluation, and application of the glyphs that are utilized in our file activity visualization tool. Through iterative design and team discussions, our understanding and appreciation of the concept improved along with this process. As a result of this process, we finalized our design for 18 glyphs that represent the most common events in a file system (see Figure 5). These events include creation, modification, deletion, copying, moving, and renaming. The action may be applied to a file, directory, device, shortcut (symbolic link), or metadata. The designs of these event glyphs evolved in several stages.

Initial Design

We first designed a set of glyphs in conjunction with the overall visual design of the visualization tool (shown on the right side of Figure 5). This allowed us to appreciate how these glyphs may be used and to identify the typical display conditions, such as glyph sizes, density, and available visual channels. At this stage, we decided that the basic glyph designs should not feature the hue channel; we reserved this intuitive and powerful visual channel to depict user- or data-specific variables.

Expert Estimation

Four visualization researchers took part in this research, and all had publications in areas of glyph-based visualization. We used our knowledge about different visual channels and our experience in glyph designs to improve on the

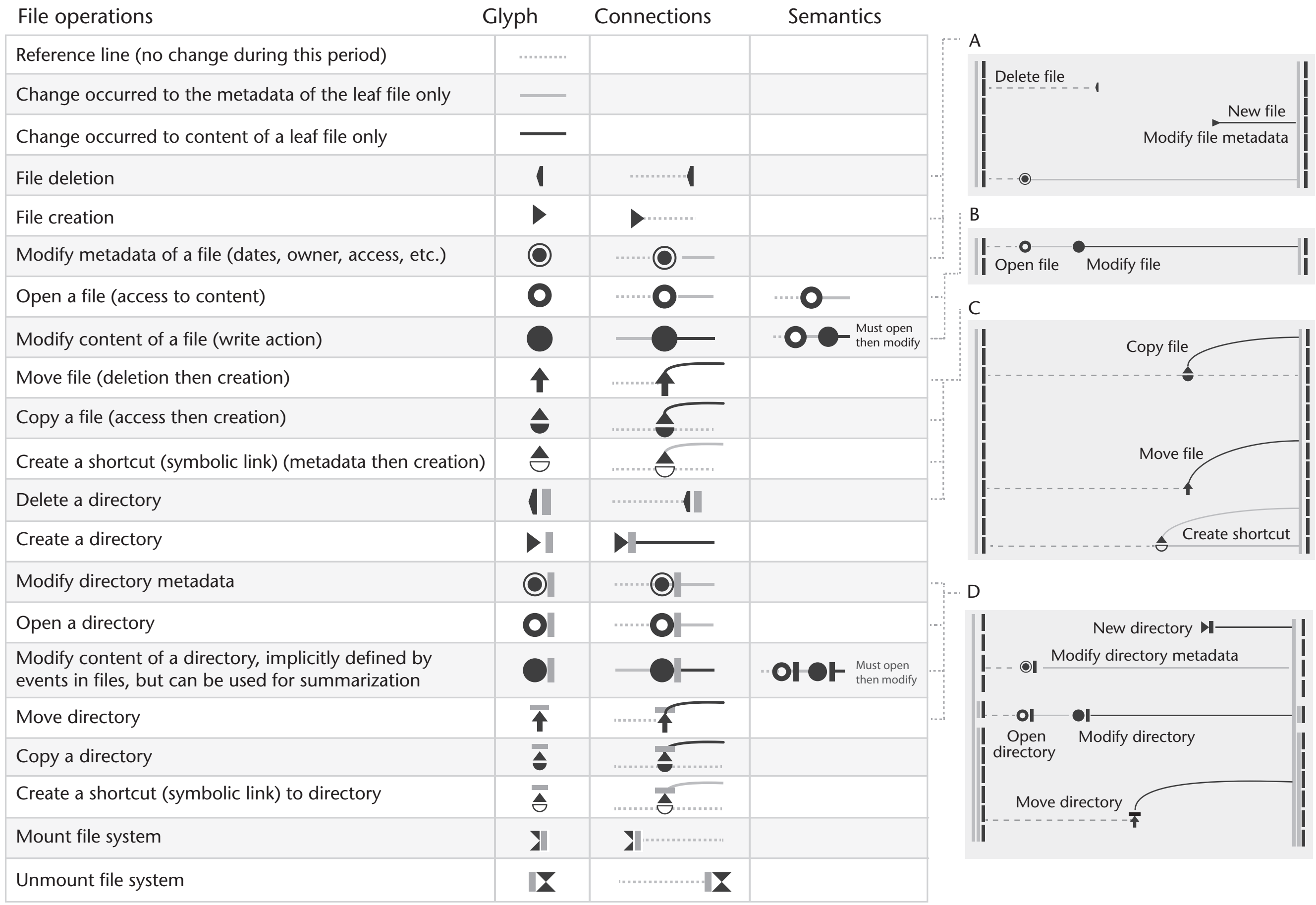

Figure 5. The 18 glyphs designed to represent different file-system events. Each event is associated with its primary glyph representation in the second column. In addition, an event may be associated with special signatures in terms of connection (third column) and semantic ordering (fourth column). The graphs on the right side (A–D) illustrate the initial designs for the timeline visualization, which depicts file activity using the proposed glyph set with a hierarchical tree representation of the file system to provide context.

original designs. This is similar to the estimation by expert designers approach (method 1) we discussed previously.

In this evaluation, we noticed that although we agreed on how easy or difficult it was to differentiate between pairs of glyphs, we could not easily agree on the reasons why. When we explicitly tried to determine the QHD between a pair of glyphs, we were often influenced by different features, such as component shapes, convexity, aspect ratio, and curvature. This experience led us to further appreciate the multifaceted complexity in estimating QHD. Many of the glyph designs in Figure 5 were stabilized at this stage.

Crush Tests

We applied crush tests to all glyphs designed during the case study (method 2). In several cases, we carried out systematic testing by applying consistent zooming factors to all glyphs. More often, when we were considering individual glyphs, we carried out ad hoc crush tests using our drawing software, such as by zooming and overlaying a translucent shape on top of glyphs.

At this stage, we realized that simulating different conditions that would cause glyph quality to degenerate was not a trivial undertaking. In many ways, this also echoed the multifaceted nature and complexity in estimating QHD that we mentioned earlier.

Human-Centric Estimation

As we discussed earlier, we conducted a survey to gain a better understanding about QHD in the context of individual visual channels. This also allowed us to evaluate the set of proposed event glyphs. We considered 20 glyph designs, for which there were 190 pairwise comparisons. We selected 48 pairs that we considered to be “more risky” than other pairs in terms of differentiability. Figure 6 shows three example questions from the questionnaire, in which we asked users how difficult or easy it is to differentiate these pairs of glyphs. In this particular example, the first two pairs are

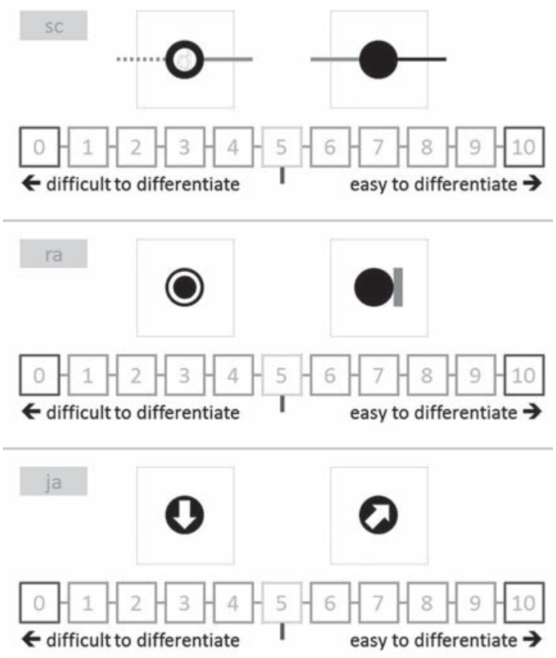

Figure 6. Three examples from the survey questionnaire. Participants were asked, "How difficult or easy is it to differentiate between these pairs of glyphs?" There were a total of 104 questions, based on 68 different glyphs.

from our proposed glyph set (groups S and R) and the third example is a primitive pair for measuring orientation (group J). The full questionnaire is available in the supplementary material.

In the survey, we found that only one pair scored below 2 bits in terms of QHD. The final designs of the glyphs did not include this pair. The details of this evaluation are provided in the "Evaluating Event Glyphs" section.

Computer-Based Similarity Measures

We used the same similarity metric we mentioned earlier to measure the QHD of the 48 pairs that were potentially risky. We found that they all passed this QHD test. The details of this evaluation are also provided in the "Evaluating Event Glyphs" section.

Deployment in Software

In addition to this design effort, we incorporated the glyph set into the visualization tool and used the tool to visualize events in a Dropbox folder. This helped us gain direct experience with these glyphs and learn how they might be viewed and interpreted in practical applications. The details of this deployment are discussed in the "Visualizing Dropbox Activity Logs" section.

Further Considerations

Differentiability is only one aspect of glyph design. We have to consider other aspects such as how easy it is to learn and remember glyphs, how glyphs may be connected, and how they may be ordered if the corresponding events happened to the same file or directory.

As Figure 5 shows, we utilized some similar designs for files and directories to assist in learning and memorization. Meanwhile, we also considered how they may be connected. The three types of connection lines are shown on the top of Figure 5, and the different orientations shown in the third column may potentially add additional features for differentiating glyphs. For example, all lines connecting to a deletion glyph will always come from the left, and all lines connecting to a creation glyph will always extend to the right. All lines connecting to a copy or move glyph will suggest a spatial shift vertically. In addition, semantic ordering, such as to open and read a file, can also increase the QHD, as Figure 5 illustrates.

Evaluating Event Glyphs

As part of the study we described earlier, we evaluated the glyphs in Figure 5 based on the QHD obtained from a human-centric survey and using computer-based similarity measures. This allowed for a comparative analysis against the reference pairs and primitive glyph sets. For our glyph set, only the potentially risky pairs were evaluated.

The human-centric estimation provided us with the most meaningful insight about the glyphs' quality. The 104 pairs of glyphs evaluated by participants received an average QHD of 2.9 bits. The average QHD for the reference pairs (groups A and B) is 2.7 bits. The average for the primitive pairs (groups C to J) is 2.6 bits. For our proposed glyph set, the average QHD for the potentially risky pairs (groups O to Z) is 3.2 bits, suggesting that these glyphs are well differentiable.

Almost all our glyph pairs for file-system visualization have a QHD above 2 bits. Only one pair received QHD = 1.5 bits, and it was not used in the final design. The upper part of Figure 7 shows a subset of the survey results. The computer-based metric also measured our glyph pairs favorably. As we mentioned previously, the average QHD estimated by the metric is normalized to have the same min, mean, and max as the human-centric estimation. The complete set of 104 glyph pairs have an average QHD of 2.9 bits. The average QHD for the reference pairs is 2.6 bits. The average for the primitive pairs is 2.7 bits. The average for the potentially risky pairs in our glyph set is 3.1 bits, again suggesting that the proposed set of glyphs are well differentiated. The lowest QHD for our potentially risks glyph pairs is 2.1 bits.

The evaluation also revealed some interesting phenomena. The additional features added to the directory glyphs reduced the QHD among the directory glyphs. For example, when comparing

Related Work in Glyph Visualization

The work described in this article covers the broad the definition of glyphs provided by Rita Borgo and her colleagues.[1] We thus consider a glyph

> a small visual object that can be used independently and constructively to depict attributes of a data record or the composition of a set of data records. Each glyph can be placed independently from others, while in some cases, glyphs can be spatially connected to convey the topological relationships between data records or geometric continuity of the underlying data space. Glyphs are a type of visual sign that can make use of visual features of other types of signs such as icons, indices, and symbols.

Matthew Ward provided a technical framework for glyph-based visualization that covers aspects of visual mapping and layout methods as well as addresses important issues such as bias in mapping and interpretation.[2,3] The state-of-the-art report on glyph-based visualization by Borgo and her colleagues also compiles many of the design guidelines and techniques that have been utilized in the field.[1] Researchers have discussed various design considerations for glyph-based visualization for 3D data, including data mapping, glyph instantiation, and rendering.[4]

Glyphs have also been used in a variety of applications. For example, Philip Legg and his colleagues proposed MatchPad for analyzing sports event data using glyph-based visualization.[5] Thomas Kapler and William Wright proposed GeoTime for displaying military events in a combined temporal and geospatial visualization.[6] J. Pearlman and P. Rheingans used glyphs to visualize network security events.[7] Martin Suntinger and his colleagues used glyph-based event visualization to create an event tunnel for business analysis and incident exploration.[8] Colin Ware and Matthew Plumlee investigated the use of glyph-based visualization for encoding multivariate weather data such as temperature, pressure, wind direction, and wind speed.[9] And Johannes Fuchs and his colleagues conducted an evaluation study that addressed temporal glyph designs for small multiple displays.[10]

References

1. R. Borgo et al. “Glyph-Based Visualization: Foundations, Design Guidelines, Techniques and Applications,” *Eurographics State of the Art Reports*, May 2013, pp. 39–63.
2. M.O. Ward, “A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization,” *Information Visualization*, vol. 1, nos. 3–4, 2002, pp. 194–210.
3. M.O. Ward, *Multivariate Data Glyphs: Principles and Practice*, Springer, 2008.
4. A.E. Lie, J. Kehrer, and H. Hauser, “Critical Design and Realization Aspects of Glyph-Based 3D Data Visualization,” *Proc. 25th Spring Conf. Computer Graphics* (SCCG), 2009, pp. 19–26.
5. P.A. Legg et al., “MatchPad: Interactive Glyph-Based Visualization for Real-Time Sports Performance Analysis,” *Computer Graphics Forum*, vol. 31, no. 3, 2012, pp. 1255–1264.
6. T. Kapler and W. Wright, “GeoTime Information Visualization,” *Proc. IEEE Information Visualization*, 2004, pp. 25–32.
7. J. Pearlman and P. Rheingans, “Visualizing Network Security Events Using Compound Glyphs from a Service-Oriented Perspective,” *Proc. Workshop on Visualization for Computer Security* (VizSEC 2007), 2008, pp. 131–146.
8. M. Suntinger et al., “The Event Tunnel: Interactive Visualization of Complex Event Streams for Business Process Pattern Analysis,” *Proc. IEEE Pacific Visualization*, 2008, pp. 111–118.
9. C. Ware and M.D. Plumlee, “Designing a Better Weather Display,” *Information Visualization*, vol. 12, 3–4, 2013, pp. 221–239.
10. J. Fuchs et al., “Evaluation of Alternative Glyph Designs for Time Series Data in a Small Multiple Setting,” *Proc. Conf. Human Factors in Computer Systems* (CHI), 2013, pp. 3237–3246.

groups O and Q, where the glyphs for creation, modify metadata, and modify content were compared within the context of files (group O) and directories (group Q), the human-centric estimation shows a noticeable difference. The average QHD for group O (files) is 3.4 bits and that for group Q (directories) is 2.6 bits. Yet, the glyphs for directories are similar to those of files, with the addition of the rectangle to the right of the circular region.

Similarly, for groups T and V, where move, copy, and shortcut glyphs were compared, the average QHD for group T (files) is 3.3 bits and for group V (directories) is 2.8 bits. Meanwhile, the computer-based similarity measures suggest little difference between groups O and Q and between groups T and V, which given the similar design is understandable. This suggests that further research is necessary to enrich the existing findings about how the distance functions for integrated and separable visual channels may affect perception.[5]

Visualizing Dropbox Activity Log

To demonstrate the applicability of the proposed glyph set, we developed an interactive tool for visualizing file event log data. The system consists of a Python backend for processing logs from file storage services such as Dropbox and Git, and a web-based frontend that provides the user interface. The frontend was created with a combination of HTML5,

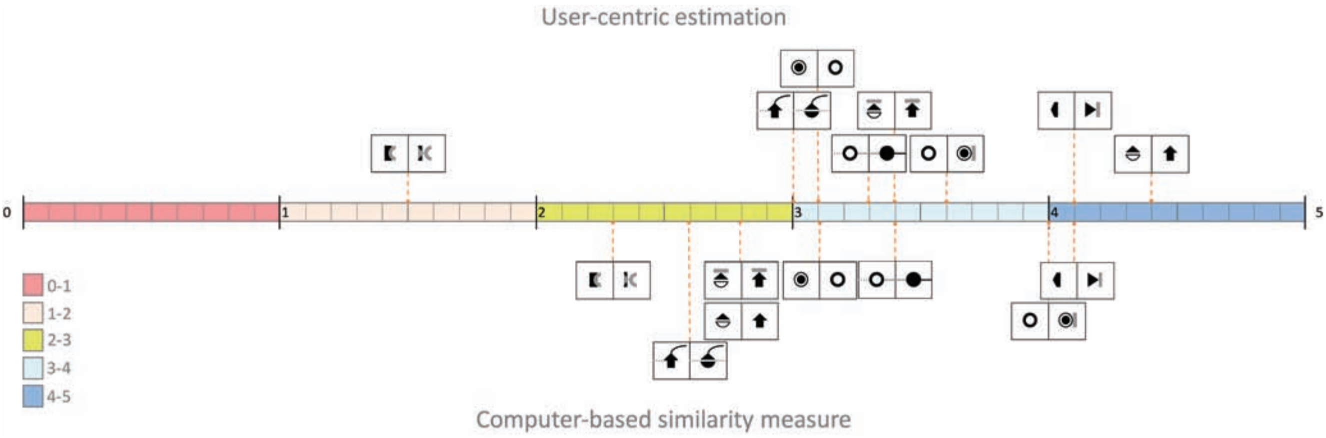

Figure 7. Sample of results for the file visualization glyph pairs used in both the user-centric estimation (top) and computer-based similarity measure (bottom). Low values indicate difficult to differentiate, and high values indicate easy to differentiate. The color legend shows the QHD ranges, where red indicates a low QHD ($0 \leq$ QHD ≤ 1) and blue a high QHD ($4 \leq$ QHD ≤ 5).

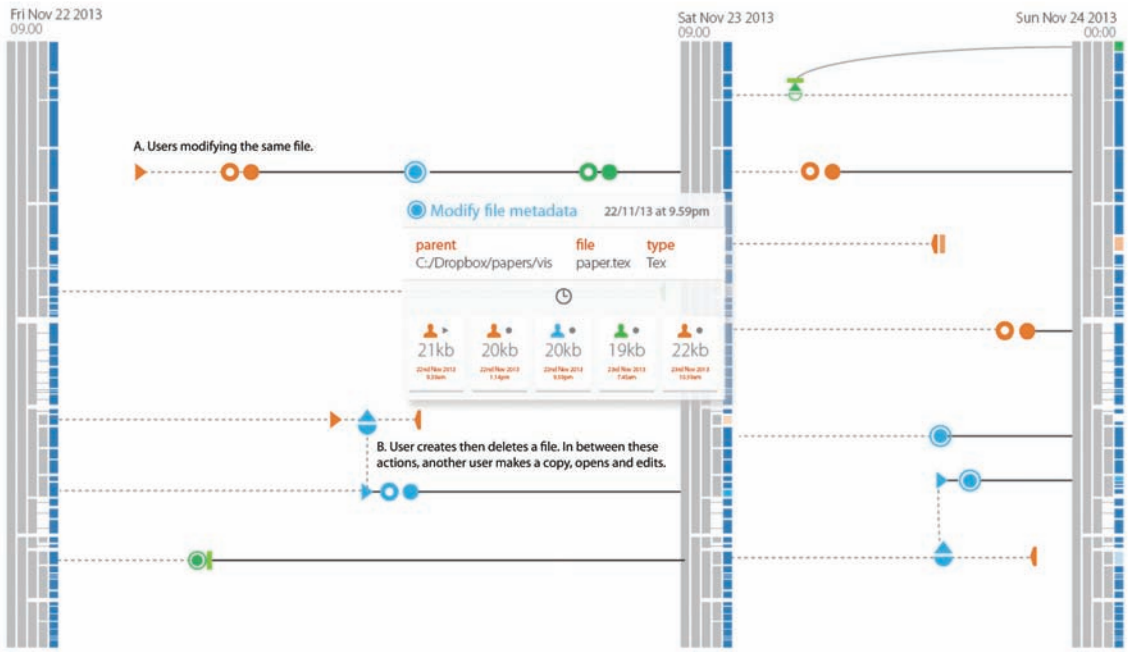

Figure 8. Glyph-based visualization displaying events in a Dropbox activity log. The vertical bars are an abstract representation of a directory tree, and the timeline flows from left to right. At point A, a number of users have modified the same file. The popup gives more details about the modifications, who made them, and when to allow for provenance tracking. Point B shows another interesting case where user X (orange) created a file; then user Y made a copy of this file, opened it, and modified it. User X then deleted the original file, but the copy user Y modified still exists elsewhere.

CSS, and JavaScript (utilizing Raphael.js for the visualization element and jQuery for control of popup events). In addition to glyph-based visualization, the system supports a variety of interactions:

- filtering different types of file-system events,
- filtering different users,
- selecting a specific directory as a subtree,
- selecting different time periods,
- zooming and scrolling on the directory axis and timeline, and
- displaying detailed view when a mouse hovers over a glyph.

Figure 8 shows an excerpt of the file-system visualization, depicting events from a Dropbox activity log. Because Dropbox supports file-sharing capabilities, it is desirable for users to visualize events in a shared folder, for instance, to see which file was created or modified recently and by whom. Although the service does provide a text-based activity log that users can access, it is time-consuming and tedious to read a long list of events. Glyph-based visualization gives users an easy overview of the events in a shared folder. The visualization in Figure 8 makes use of the glyph set shown in Figure 5, with color used to depict different users. This example includes three different users (blue, orange, and green) who have accessed the system during this time.

The glyphs are shown in conjunction with the file system hierarchy, which is represented at each of the three time intervals by the gray vertical bars.

We can see that some files were accessed by all three users, such as the top line between the first and second time steps. Here, the orange user created a file and then opened and modified it. Later, the blue user modified the file's metadata (possibly renaming it or changing its access date). Afterward, the green user opened and modified the file. By hovering over each glyph with the mouse, the user can display a popup window that provides further detail about the activities, including the sequence of file events and the associated file size. This enables collaborative colleagues to easily understand ongoing actions, reducing the need to update other users manually.

We consider this to be the first step toward establishing a collection of mathematical and cognitive theories, experimental findings and statistics, design techniques, and computational metrics for guiding and aiding glyph designs. This work highlights a number of gaps where further research is needed. For example, it would be highly desirable to understand the relationship between the JND measures of various visual channels and differentiability of glyphs encoded using such visual channels. It will also be beneficial to correlate existing findings about error detection and correction with QHD measures.[7] In the future, to further aid glyph design processes, it will be necessary to develop computer-based similarity measures that are statistically closer to (or even better than) human-centric estimation.

References

1. M. Chen and H. Jänicke, "An Information-Theoretic Framework for Visualization," *IEEE Trans. Visualization and Computer Graphics*, vol. 16, no. 3, 2010, pp. 1206-1215.
2. R.W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Technical J.*, vol. 29, no. 2, 1950, pp. 147–160.
3. M. Chen et al., "Visual Multiplexing," *Computer Graphics Forum*, vol. 33, no. 3, 2014, pp. 241–250.
4. P. Rheingans and C. Landreth, "Perceptual Principles for Effective Visualizations," *Perceptual Issues in Visualization*, Springer-Verlag, 1995, pp. 59–74.
5. E. Maguire et al., "Taxonomy-Based Glyph Design with a Case Study on Visualizing Workflows of Biological Experiments," *IEEE Trans. Visualization and Computer Graphics*, vol. 18, no. 12, 2012, pp. 2603–2612.
6. C. Zitova and J. Flusser, "Image Registration Methods: A Survey," *Image and Vision Computing*, vol. 21, no. 11, 2003, pp. 977–1000.
7. M.-K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Trans. Information Theory*, IT-8, 1962, pp. 179-187.
8. B. Johnson and B. Shneiderman, "Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures," *Proc. IEEE Visualization*, 1991, pp. 284–291.

Philip A. Legg *is a senior lecturer in computer science at the University of the West of England. His research interests include visualization, machine learning, and human-computer interaction. Legg has a PhD in computer science from Cardiff University. Contact him at phil.legg@uwe.ac.uk.*

Eamonn Maguire *is a research scientist at the CERN European Laboratory of Particle Physics. His research interests include visualization and machine learning. Maguire has a DPhil in computer science from the University of Oxford. Contact him at eamonnmaguire@gmail.com.*

Simon Walton *is a research associate in the e-Research Center at the University of Oxford. His research interests include visualization and computer graphics. Walton has a PhD in computer science from Swansea University. Contact him at simon.walton@oerc.ox.ac.uk.*

Min Chen *is a professor of scientific visualization in the e-Research Center at the University of Oxford. His research interests include visualization, computer graphics, and human-computer interaction. Chen has a PhD in computer-aided engineering design from Swansea University. Contact him at min.chen@oerc.ox.ac.uk.*