

Glyph Visualization – A fail-safe design scheme based on quasi-Hamming distances

Philip A. Legg University of the West of England, UK.

Eamonn Maguire CERN European Laboratory for Particle Physics.

Simon Walton University of Oxford, UK.

Min Chen University of Oxford, UK.

In many applications of spatial or temporal visualization, glyphs provide an effective means for encoding multivariate data. Glyph-based visualizations are ubiquitous in modern life since they make excellent use of the human ability to learn abstract and metaphoric representations to facilitate 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 in order to facilitate mapping consistency, semantic interpretation, learning and memorization. As shown in Figure 1, as the number of glyphs increase and the visualization scale decreases, in many cases this will often result in glyphs becoming indistinguishable. Similarly, color defects from poor printing or uncalibrated display screens could also lead to difficulties in glyph differentiation.

Whilst many glyph designers will often aim to account for perception and legibility in their work, they may also wish to incorporate their own creativity and intuition as an artist, and so there is a delicate balance to be addressed

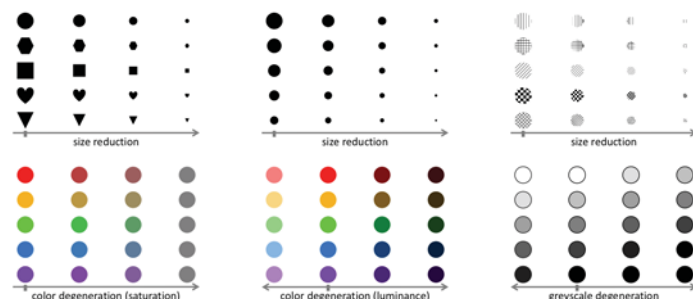


Figure 1: Different types of quality degeneration are applied to several glyphs, each of which is encoded using a single visual channel. The original quality is indicated by a marker on the x-axis. When size, saturation and luminance are changed, they become more difficult to differentiate.

between the science and the art of effective glyph design. This poses some challenging research questions for glyph visualization, such as, “*Is there a theoretical framework to encompass various design guidelines?*” and “*Is there a systematic approach to design a fail-safe glyph set?*” While this work is a direct attempt to answer the second question, it also makes the connection between glyph-based visualization with information theory, which has been considered as a candidate framework for visualization [Chen10].

A Background on Glyph Visualization

Borgo et al. [Borgo13] provide a narrow and a broad definition of glyphs. The work described in this paper covers the broad scope given [Borgo13], we hereby consider a glyph as ‘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.’

Ward [Ward02, Ward08] provides a technical framework for glyph-based visualization that covers aspects of visual mapping and layout methods, as well as addressing important issues such as bias in mapping and interpretation. The state of the art report on glyph-based visualization by Borgo et al. [Borgo13] also compiles many of the design guidelines and techniques that have been utilized in the field. Lie et al. [Lie09] discuss a variety of design considerations for glyph-based visualization including data mapping, glyph instantiation, and rendering, for three-dimensional data. Glyphs have been used in a variety of different applications. For example, Legg et al. [Legg12] propose *MatchPad* for analysing sports event data using glyph-based visualization. Kapler and Wright [Kapler04] propose *GeoTime* for displaying military events in a combined temporal and geo-spatial visualization. Pearlman and Rheingans [Pearlman08] use glyphs for visualizing network security events. Suntinger et al. [Suntinger08] use glyph-based event visualization to create an *Event Tunnel* for business analysis and incident exploration. Ware and Plumlee [Ware13] investigate the use of glyph-based visualization for encoding multi-variate weather data such as temperature, pressure, wind direction and wind speed. Fuchs et al. [Fuchs13] also conducted an evaluation study that addressed temporal glyph designs for small multiple displays.

[Borgo13] BORGIO R., KEHRER J., CHUNG D. H., MAGUIRE E., LARAMEE R. S., HAUSER H., WARD M., CHEN M.: Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *Eurographics State of the Art Reports* (May 2013), EG STARS, pp. 39–63.

[Fuchs13] FUCHS J., FISCHER F., MANSMANN F., BERTINI E., ISENBERG P.: Evaluation of alternative glyph designs for time series data in a small multiple setting. In *Proc. Conference on Human Factors in Computer Systems (CHI)*, (2013), pp. 3237–3246.

[Kapler04] KAPLER T., WRIGHT W.: Geotime information visualization. In *Proc. IEEE Information Visualization* (2004), pp. 25–32.

[Legg12] LEGG P. A., CHUNG D. H. S., PARRY M. L., JONES M. W., LONG R., GRIFFITHS I. W., CHEN M.: MatchPad: Interactive glyph-based visualization for real-time sports performance analysis. *Computer Graphics Forum* 31, 3 (2012), 1255–1264.

[Lie09] LIE A. E., KEHRER J., HAUSER H.: Critical design and realization aspects of glyph-based 3D data visualization. In *Proc. SCCG* (2009), pp. 27–34.

[Pearlman08] PEARLMAN J., RHEINGANS P.: Visualizing network security events using compound glyphs from a service-oriented perspective. In *Proc. VizSEC 2007*, Springer Mathematics and Visualization. 2008, pp. 131–146.

[Suntinger08] SUNTINGER M., SCHIEFER J., OBWEGER H., GROLLER M. E.: The event tunnel: Interactive visualization of complex event streams for business process pattern analysis. In *Proc. IEEE Pacific Visualization* (2008), pp. 111–118.

[Ward02] WARD M. O.: A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization* 1, 3/4 (2002), 194–210.

[Ward08] WARD M. O.: *Multivariate Data Glyphs: Principles and Practice*. Springer Handbooks Comp. Statistics. 2008.

[Ware13] WARE C., PLUMLEE M. D.: Designing a better weather display. *Information Visualization* 12, 3-4 (2013), 221–239.

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. In particular, we introduce the concept of a **quasi-Hamming distance** in the context of glyph design. We examine the feasibility of estimating 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 a file system event visualization that can depict the activities of multiple users. 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 that this new concept will encourage designers to consider systematically the need for empowering visualization users to detect and correct potential communication errors.

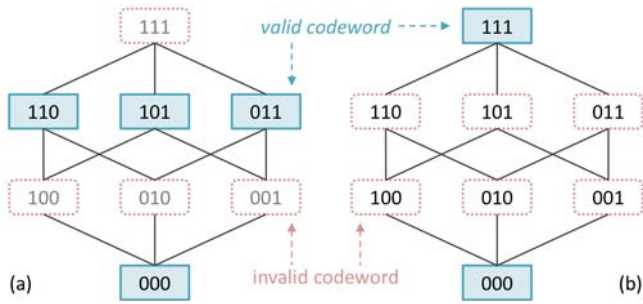


Figure 2: Two 3-bit codes. (a) A code can detect 1-bit errors. (b) A code can detect 2-bit errors and correct 1-bit errors.

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 [Hamming50] is a measure of the number of bit positions in which two codewords differ. Considering all pairs of codewords in a code, the minimal distance is referred to as the minimal Hamming distance of the code. (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** allows the receiver to discover that any error has occurred and to 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:

- **Theorem.** 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 [Hamming50].

For example, given a 3-bit code as illustrated in Figure 2 there are 8 possible codewords. One may select a subset of these codewords to construct a code with its minimal Hamming distance equal to 2 bits or 3 bits. Figure 2(a) shows one of such codes, which has 4 codewords and is of 2 bits Hamming distance. This code can detect 1-bit errors since any change of a valid



Figure 3: Two examples that illustrate the phenomena of error detection and error correction in glyph-based visualization. In the first example, the orientation of the glyph on the left is distorted yet a viewer may sense this and consult the legend for correction. 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, though the top point of the star is missing because of occlusion.

codeword by 1 bit would result in an invalid codeword, which would lead the receiver to discover the error. Figure 2(b) shows another code with 2 codewords, and of 3 bits Hamming distance. It can detect 2-bit errors and correct 1-bit errors. When a valid codeword (e.g., 111) is changed by 1 bit during transmission (e.g., 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 make a correct 'correction'. 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 quite 'right', conscious or unconscious effort can be made to correct such an error. Conscious effort, which is an analogy of error detection and repeated transmission, may include zooming in to have a close look, or consulting the legend. Unconscious effort, which is an analogy of error correction, can be the result of gestalt effects (the nearest neighbor) [Chen14], or a combined judgement involving multiple visual components (redundancy) [Rheingans95].

Figure 3 shows two example glyph sets, each with 8 codewords. Given the two display errors depicted on the left, i.e., an arrow glyph is skewed and a shape glyph is occluded by another shape, one 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 error 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 we 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 as a kind of measurement of "perceptual distance" between two glyphs. The introduction of QHD brings several benefits. (i) It connects glyph-based visualization to information theory through an important and widely-used concept (i.e., Hamming distance) in computing and communication. (ii) It relates the need for accurate perception of glyphs to error detection and correction by the receiver. (iii) It facilitates quantitative measures semantically equivalent to Hamming distance and its mathematical implication.

The main research questions are thereby:

- whether we can establish a measurement unit common to both measures, and
- how we can obtain such measurements.

For the first question, we can utilize 'bit' as the 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, \dots, 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 codewords c_i and c_j as $|i-j|$ bits. If c_i is mistaken for c_j , we can call this a $d=|i-j|$ -bit error where $d=|i-j|$. Now let us extend this concept to a less ordered visual channel (e.g., hue) or an integrated channel (e.g., color). Theoretically, we can construct a code C by uniformly sampling the space of the

visual channel (e.g., the CIE $L^*a^*b^*$ color space) while ensuring that every pair of samples differ by at least the JND of this channel. These codewords, i.e., samples, can be organized into a network, where the distance between any two codewords can be approximated proportionally according to the JND (i.e., $JND = 1$ bit). Note that the possible perception error rate with a code that maximizes the number of codewords based on JND is likely to be very high. In practice, one designs a glyph set 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, though in a longer term, JND can provide an *absolute reference measure* once we have obtained such measures for most visual channels in visualization.

This leads to the second research question, i.e., given a glyph set, how can we measure the distance between glyphs? One may 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. While this approach may be most convenient to the designers, its effectiveness depends very much on the experience of the designers concerned and it is rather easy to overlook certain types of display and perception errors.
2. **Crush tests.** Introduced by Maguire *et al.* [Maguire12], crush tests rescale glyphs to lower pixel resolutions, to assess the preserved detail. One 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 defined as QHD. While this approach would yield more consistent estimation of QHD, more research would be required to compile a list of different causes of errors and define coherent levels of degeneration.

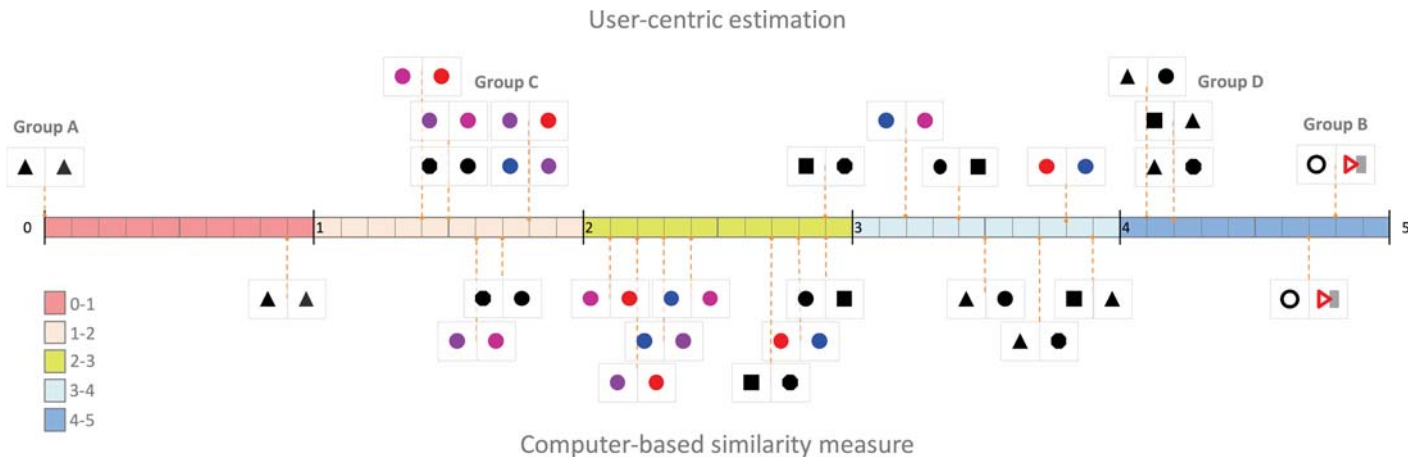


Figure 4: Sample of results for the primitive glyph pairs used in both the user-centric estimation (top) and by computer-based similarity measure (bottom). Low values indicate difficult to differentiate and high values indicate easy to differentiate. Examples from Groups A-D are highlighted, where A) hard to differentiate, B) easy to differentiate, C) differentiate by color, and D) differentiate by shape. The colour legend shows the QHD ranges. Red shows a low QHD ($0 \leq \text{QHD} \leq 1$) and blue shows a high QHD ($4 \leq \text{QHD} \leq 5$).

3. **Task-based evaluation.** Similar to (2), one can simulate different visualization conditions, enlist users to perform their tasks, measure users' 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.** One may conduct a survey among human participants about how easy or difficult it is to differentiate different glyphs. By removing task-dependency in (3), more participants can be involved in such a survey, yielding more reliable estimation of QHD.
5. **Computer-based similarity measures.** There are a variety of image similarity measures already in the literature [Zitova03]. In a longer term, it is likely that we will be able to find measures that are statistically close to user-centric estimation, though there is not yet a conclusive confirmation about optimal image similarity measures, and there are hardly any metrics specially designed for measuring similarity of glyphs.

To demonstrate the feasibility of estimating QHD, we conducted two proof-of-concept experiments based on methods (4) and (5). We did consider a task-based evaluation (method 3) as a possible approach, since the glyph set under consideration

is for a file activity visualization. However, this additional cognitive load may distort the assessment of glyph similarity. Instead, for the purpose of this current study, method 4 allowed for a much more general audience to participate in the study, and method 5 allows us to corroborate the two methods chosen. We conducted a survey among 20 participants, all of whom are either employees or students at the University of Oxford. About 50% of participants had encountered glyph-based visualization previously. The results of one participant were considered as an outlier and were not included in the statistics. After a brief introduction by one co-author of this paper, the participants were asked 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, 8 *reference pairs*, 48 *primitive pairs*, and 48 *application-specific pairs*. The 8 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 (i.e., minimal distance). In another four pairs, the two glyphs can be differentiated with undisputable ease (i.e., maximal distance).

The primitive pairs are divided into 8 groups, namely hue, shape, components, connection lines, luminance, size, texture, and orientation. In each group, different pairs feature glyphs with different perceptual distances, allowing us to obtain the human-centric estimation of the QHD

for basic visual channels individually. The application pairs contain glyphs designed for our application case study. We will discuss these in detail in the Case Study section.

The 96 primitive and application-specific pairs were mixed together in a randomized order. The 8 reference pairs were placed at positions 1, 2, 35, 36, 69, 70, 103 and 104 for helping participants to regularize their scores and for enabling us to check temporal consistency. The details about the questionnaire, the stimuli grouping and the survey results can be found in the supplementary materials. Here we briefly describe the survey results in relation to the reference and primitive pairs.

As mentioned early, we divided the category of reference pairs into two groups. Group A consists of 4 pairs of very similar glyphs, and Group B consists of 4 pairs of very different glyphs. We expected that participants will assign very low scores (difficult to differentiate) to those in A and high scores (easy to differentiate) to those in B respectively. We place one pair from A and one from B at regular intervals. The average scores for the 4 pairs in group A are (0.0, 0.4, 1.4 and 2.8) respectively and those for group B are (9.3, 9.0, 8.9, 9.7) respectively, indicating that they have statistically served as references for the minimal and maximum QHD in this survey.

The category of primitive pairs consists of 8 groups (C-J) for estimating QHD in relation to 8 visual channels (C: hue, D: shape, E: components, F: connection lines, G: luminance, H: size, I: texture, J: orientation). A sample of the groups can be seen in Figure 4, with full group details available in the supplementary material. Each group has 6 pairs of stimuli, facilitating pairwise comparison of 4 different codewords of each channel. For hue and luminance channels, after choosing the 1st and 4th codewords we used a perceptually uniform colour model (Hunter's Lab) to determine the 2nd and 3rd codewords at 50% and 75% distance from the 1st. The upper part of Figure 4 shows a small selection of survey results, where we converted the [0, 10] score range to a [0, 5] QHD range. We consider a QHD <2 as potentially risky for error detection, and a QHD <3 as potentially risky for error correction.

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 (Colour)	2.2	1.5
H (Size)	2.1	4.2
G (Luminance)	1.2	2.4

Table 1: Average QHD results for the primitive glyph pairs by group. They were obtained from the estimation by human participants and from the algorithmic similarity measure by computer. They are ordered according to 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 [Hu62, Zitova03], a well-established approach in Computer Vision that is widely-used for image similarity. The metric incorporates both pixel colour and spatial occupancy to assess the difference between the 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 $\leq 80\%$ 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, 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, shown by group. Based on the human participants, shape, component and orientation all score QHD ≥ 3 , which suggests they are well separable visual channels, however luminance was the only group to score below QHD <2, suggesting this is not well separable. Whilst some visual channels also score similarly by the computer-based similarity measure (e.g., shape, component, connection), other channels

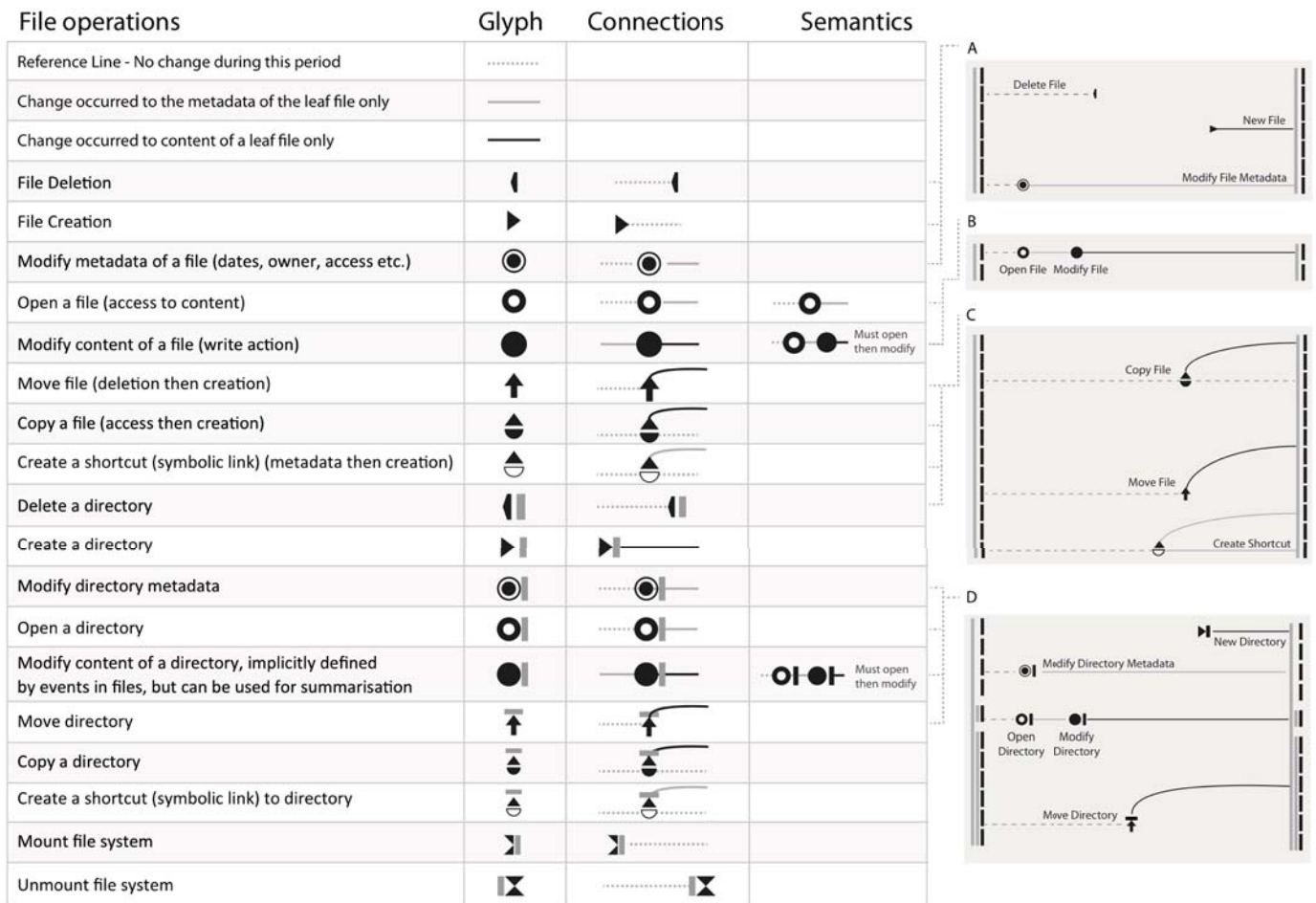


Figure 5: The 18 glyphs designed to represent different events in file systems. Each event is associated with its primary glyph representation in the second column. In addition, an event may be associated with special signatures in terms of connection (in the third column) and semantic ordering (in the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

differ from the user feedback (e.g., size, orientation). This shows that there is certainly further research to be conducted in this area, of 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, Johnson and Shneiderman, who were motivated by the need to visualizing the structure of a file system, published their seminal paper on treemaps [Johnson91]. Today, not only are file systems much larger and contain many more files, they are also shared by many more users and have many more events. One important aspect of a file system is to support collaborative activities, such as sharing files within multi-partner projects and developing software by a team of

programmers. While 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 for allowing 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 are several technical challenges. Firstly, the hierarchical nature of the file system needs to be depicted so that the spatial context of where a particular event has occurred can be identified. Secondly, the temporal information about events needs to be conveyed so that the activity ordering can also be observed and reasoned. Thirdly, there are a wide range of activities (e.g., copying a file, modifying a file) that are typically performed,

which 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, the design of glyphs 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, though the data is expected to be filtered in some way, such as for a specific portion of the file system, some specific file types, some specific user groups, etc. Nevertheless, the glyphs are expected to be relatively small, and plentiful on a display screen. The tasks of visualization 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 motivation, ability, and time to familiarize themselves with multivariate glyphs, though any metaphoric encoding will benefit learning and memorizing glyphs.

The concept of QHD has been considered throughout the design, development, evaluation and application of the glyphs that are utilized in our file activity visualization tool. Through iterative design, and discussion as a team, our understanding and appreciation of the concept have 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 (shown in Figure 5). These events include creation, modification, deletion, copying, moving and renaming. The action may be applied to a file, a directory, a device, a shortcut (symbolic link), or meta-data. The designs of these event glyphs were 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 what are the typical display conditions such as glyph sizes, density, and available visual channels. It was at this stage when we decided that the basic glyph designs should not feature the hue channel, and reserve this intuitive and powerful visual channel to depict user-specific 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 the original designs. This is similar to the Estimation by Expert designers method discussed previously. We noticed that although we could reach agreement as to how easy or difficult it can be to differentiate 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 many different features, such as component shapes, convexity, aspect ratio, and curvature. This experience led us to further appreciate the multi-faceted complexity in estimating QHD. Many of the glyph designs in Figure 5 became stabilized at this stage.

Crush Tests. We applied crush tests to all glyphs designed during the case study. 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 by using facilities in our drawing software, such as 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 multi-faceted nature and complexity in estimating QHD as mentioned above.

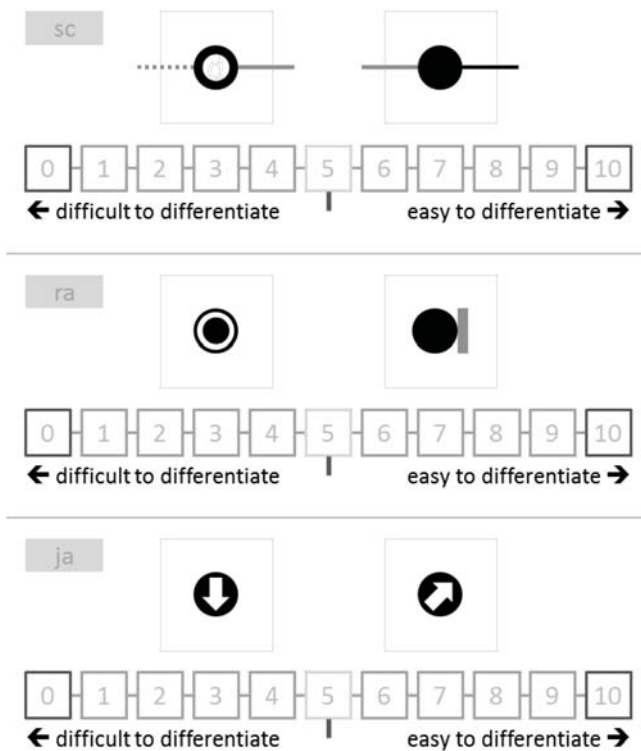


Figure 6: 3 examples from the questionnaire. Participants were asked "How difficult or easy is it to differentiate these pairs of glyphs?". In total there were 104 questions, based on 68 different glyphs.

Human-centric Estimation. As discussed in the previous Section, we conducted a survey that was designed to gain a better understanding about QHD in the context of individual visual channels, but also allowed us to evaluate the set of proposed event glyphs. We considered 20 different glyph designs, for which there would be 190 pairwise comparisons. We selected 48 pairs that were considered to be 'more risky' than other pairs in terms of differentiability. Figure 6 shows 3 example questions from the questionnaire, where users were asked how difficult or easy it is to differentiate these pairs of glyphs. In this particular example, the first two pairs are 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 1 pair scored below 2 bits in terms of QHD in the survey. The final designs of the glyphs did not include this pair. The details of this evaluation are given in the following section (Evaluating Event Glyphs).

Computer-based Similarity Measures. We used the same similarity metric as mentioned previously to measure the QHD of the 48 pairs

that might be potentially risky. We found that they all passed this QHD test. The details of this evaluation will be also given the Evaluating Event Glyphs section.

Deployment in Software. In addition to the above design effort based on the concept of QHD, we incorporated the glyph set into the visualization tool and used the tool to visualize events in a Dropbox folder. This allowed us to gain direct experience about how these glyphs might be viewed and interpreted in practical applications. The details of this deployment will be discussed in Visualizing Dropbox Activity Logs section.

Differentiability is only one aspect of glyph design. We have to consider other aspects such as how easy it is to learn and to 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 shown in Figure 5 we utilized some similar designs for files and directories to assist in learning and memorization. Meanwhile, we also consider how they may be connected. The three types of connection lines as shown on the top of the figure and the different orientations as 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 left, and all connecting to a *creation* glyph will always extend towards right. All lines connecting to a *copy* or *move* glyph will suggest a spatial shift vertically. In addition, semantic ordering, such as to *open* a file and then *read* the file, can also increase the QHD, as illustrated in Figure 5.

Evaluating Event Glyphs

We evaluated the glyphs in Figure 5 based on the QHD obtained from a human-centric survey and by using computer-based similarity measures. This was performed in the same study as described previously, and meant that 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.

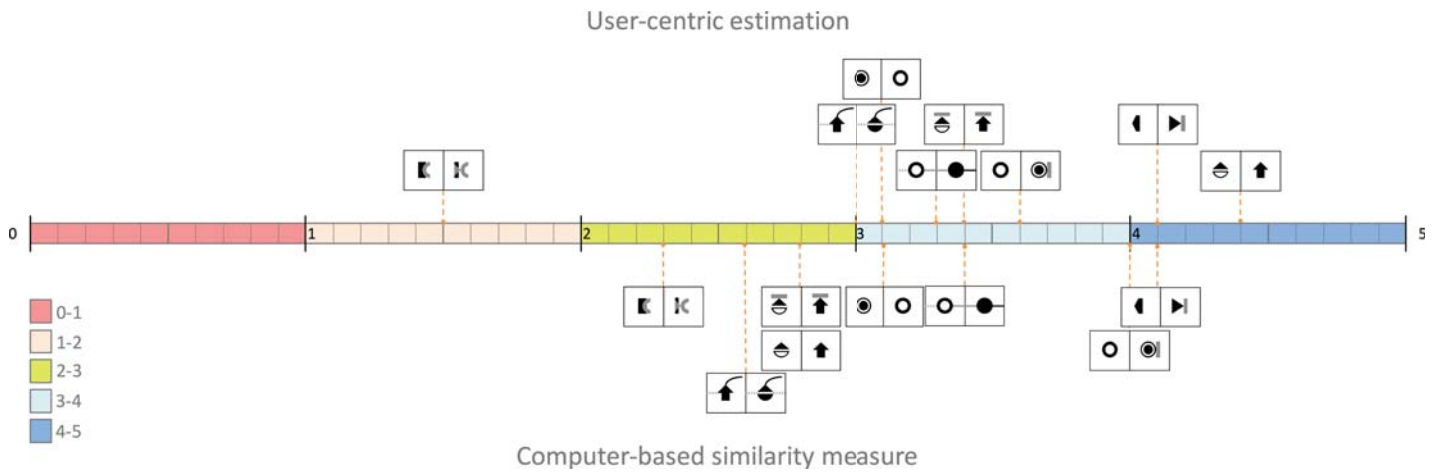


Figure 7: Sample of results for the file visualization glyph pairs used in both the user-centric estimation (top) and by computer-based similarity measure (bottom). Low values indicate difficult to differentiate and high values indicate easy to differentiate. The color legend shows the QHD ranges. Red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

The human-centric estimation provided us with most meaningful insight about the quality of the glyphs. The 104 pairs of glyphs evaluated by participants have 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 of our glyph pairs have their QHD above 2 bits, except one pair (QHD = 1.5 bits) which was not used in the final design. The upper part of Figure 7 shows a subset of the survey results. Meanwhile, the computer-based metric also measured our glyph pairs favorably. As 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 differentiate. 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 have reduced QHD among directory glyphs. For example, when comparing Group O and Group Q, where the glyphs for *creation*, *modify metadata* and *modify content*

were compared within the context of files (in Group O) and directories (in Group Q), 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 Group T and Group V where *move*, *copy* and *short cut* glyphs were compared, the average QHD for Group T (files) is 3.3 bits, and that for Group V (directories) is 2.8 bits. Meanwhile, the computer-based similarity measures suggest little difference between O and Q and between 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 [Maguire12].

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 comprises of a Python back-end for processing logs from file storage services such as Dropbox and Git, and a web-based front-end that provides the user interface. The front-end was created with a combination of HTML5, 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 including:

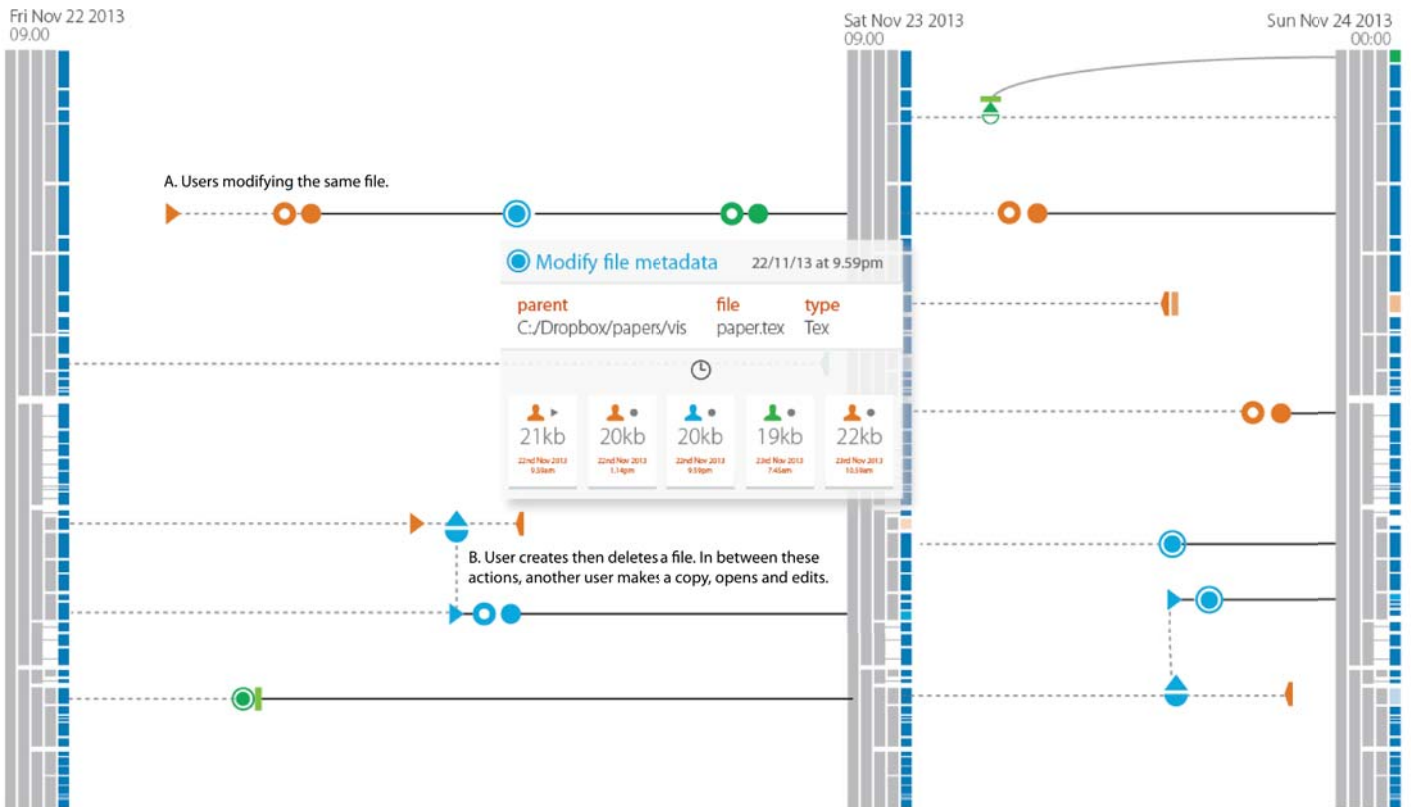


Figure 8: Glyph-based visualization is used to display events in a Dropbox activity log. The vertical bars are abstract representation of a directory tree and the timeline flows from left to right. At (A) we can see the case where a number of users have been modifying the same file. The popup gives more details about the modifications, who did them, and when to allow for provenance tracking. At (B) we have another interesting case where user X (orange) creates a file, then user Y creates a copy of this file, opens it and modifies its contents. User X then deletes the original file however a modified copy exists elsewhere.

- Filtering different types of file system events;
- Filtering different users;
- Selecting a specific directory as a subtree;
- Selecting different time period;
- Zooming and scrolling on the directory axis and timeline;
- Detailed view displayed when mouse hovers over glyph.

Figure 8 shows an excerpt of the file system visualization, depicting events from a collaborative Dropbox. Since Dropbox supports file sharing capability, it is desirable for users to visualize events in a shared folder, for instance, to see which file has been 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 to some extent, tedious to read a long list of events. Glyph-based visualization allows users to gain an overview of the events in a shared folder at ease. Here, we make use of the glyph set shown in Figure 5, with color used to depict different users. In this example, there are three different users (blue,

orange, and green) who have accessed the system during this time.

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

It can be seen that some files are 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 metadata of the file (possibly renaming it or changing its access date). Afterwards, the green user opened and modified the file. By hovering the mouse over each glyph, a pop-up window is also displayed that provides further detail of the activities, including the sequence of file events and the associated file size. This enables collaborative colleagues to understand ongoing actions much easier, reducing the need for updating other users manually.

Conclusion

We have presented a novel conceptual framework, called **quasi-Hamming distance** (QHD), which facilitates a systematic approach to designing fail-safe glyph encoding schemes. To demonstrate the feasibility of this work, we presented two proof-of-concept experiments, where we obtained QHD measures from a human-centric survey and from computer-based similarity measures. We also conducted a practical case study where Dropbox event logs were visualized using a fail-safe glyph set. From the outset, our approach was to ensure a high-level of differentiability in the glyph design, whilst achieving requirements such as minimal use of color and metaphoric consistency.

We very much consider this as the first step towards the establishment of 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 is highly desirable for us to understand the relationship between the JND measures of various visual channels and differentiability of glyphs encoded using such visual channels. It will be beneficial to correlate existing findings about error detection and correction, such as [Rheingans95], with QHD measures. It is also highly desirable to research into computer-based similarity measures that are statistically closer to (or even better than) human-centric estimation to further aid glyph design processes.

References

[Chen10] CHEN M., JÄNICKE H.: An Information-theoretic Framework for Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16, 3 (2010), 1206-1215.

[Hamming50] HAMMING R. W.: Error detecting and error correcting codes. *Bell System Technical Journal* 29, 2 (1950), 147–160.

[Chen14] CHEN M., WALTON S., BERGER K., THIYAGALINGAM J., DUFFY B., FANG H., HOLLOWAY C., TREFETHEN A. E.: Visual

multiplexing. *Computer Graphics Forum* 33, 3 (2014), 241–250.

[Hu62] HU M.-K.: Visual pattern recognition by moment invariants. *IRE Trans. On Information Theory, IT-8 (1962)*, pp. 179-187.

[Johnson91] JOHNSON B., SHNEIDERMAN B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proc. IEEE Visualization (1991)*, pp. 284–291.

[Maguire12] MAGUIRE E., ROCCA-SERRA P., SANSONE S.-A., DAVIES J., CHEN M.: Taxonomy-based glyph design with a case study on visualizing workflows of biological experiments. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2603–2612.

[Rheingans95] RHEINGANS P., LANDRETH C.: Perceptual principles for effective visualizations. In *Perceptual Issues in Visualization*. Springer-Verlag, 1995, pp. 59–74.

[Zitova03] ZITOVA C., FLUSSER J.: Image registration methods: a survey. *Image and Vision Computing* 21, 11 (2003), 977–1000.

Philip A. Legg is a Senior Lecturer in Computer Science at the University of the West of England, UK.

Eamonn Maguire is a Research Scientist at the CERN European Laboratory of Particle Physics, Switzerland.

Simon Walton is a Research Associate at the e-Research Centre, University of Oxford, UK.

Min Chen is a Professor of Scientific Visualization at the e-Research Centre, University of Oxford, UK.