EG UK Computer Graphics & Visual Computing (2014)
Rita Borgo, Wen Tang (Editors)

A Study on Glyph-based Visualisation with Dense Visual Context

Saiful Khan,[1,2] Karl J. Proctor,[2] Simon Walton,[2] René Bañares-Alcántara,[1] and Min Chen[2]

[1]Department of Engineering Science, [2]Oxford e-Research Centre, University of Oxford, UK

Abstract

In a focus+context visualization, one often finds it difficult to overlay focus information on top of a dense visual context. This work is motivated by a need for visualizing search results (i.e., focus) in relation to a treemap representation of a large file system (i.e., context). We thus consider that the focus consists of a collection of visual objects discretely-distributed over a background featuring dense context information. The conventional approach for showing such objects in focus is to use colored dots, which can encode limited information and may be difficult to discern from the context background. In this paper, we report a study on three alternative approaches, namely (a) cyclically-animated dot, (b) static glyph and (c) cyclically-animated glyph. We conducted a focus group study for qualitative evaluation and found that cyclically-animated dots and static glyphs are the preferred alternatives. While fine tuning cyclic animation is feasible, cyclically-animated glyphs do not offer an attractive solution in general.

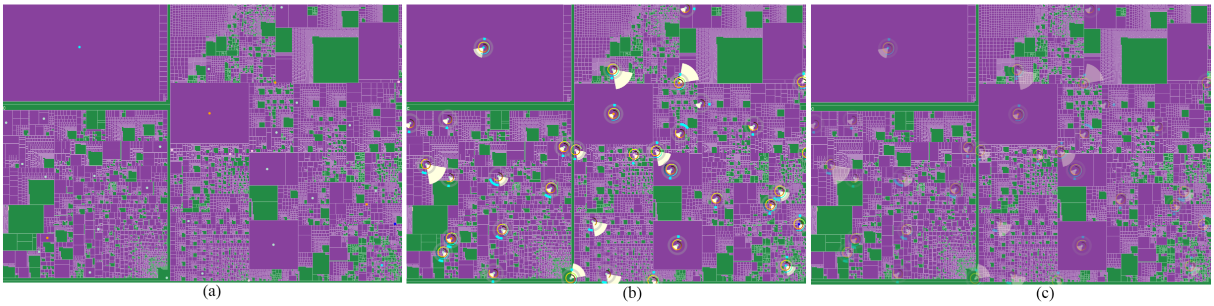
1. Introduction

This paper is concerned with *focus+context visualization*. In particular we address a requirement for displaying a collection of small objects (as the focus) discretely distributed on top of a dense visual representation (as the context). Many visual representations, such as treemaps [JS91], pixel-based visualizations [Kei00] and images, make highly cost-effective use of space in conveying complex information. However, when these representations are used to depict contextual information in a focus+context visualization, it is difficult to overlay focal information on top of the context while facilitating an efficient visual search.

For example, Fig. 1 shows a zoomed-in view of a treemap that represents part of a GNU/Linux file system. In many situations, it is useful to display search results (i.e., *focus*) in conjunction with their distribution in the file system (i.e., *context*). Such a visualization can help users determine if search criteria were set correctly or if the search has resulted in any unexpected outcomes. From Fig. 1(a), we can observe that when the search results are displayed as static colored dots against a treemap background, it is not easy to identify these dots. With the rapid growth of storage capacity due to many factors (e.g., low cost disks, distributed file systems, data repositories), the level of difficulties in, and the required effort for, searching files will likely increase. It is therefore becoming highly desirable to provide a search interface with effective focus+context visualization.

In this paper, we report a multi-faceted study on several visualization solutions for displaying a collection of small objects discretely distributed on top of a dense visual representation. In particular, we have concentrated on the scenario of searching within a large file system, partly because the study was motivated by a software engineering project for developing a search tool for a large scale distributed file system, and partly because file searching provides empirical studies with an intuitive case study that most, if not all, participants can appreciate. Perceptually, discerning dots from a treemap is challenging due to many factors, including dot size, change blindness, and selective and divided attention [BdHV94, SR05, Ren02]. Hence, one may consider several alternatives to the static dots in Fig. 1(a), e.g., using large dots or animating the dots. The approach of increasing dot size provides a further option for encoding more file attributes using *multivariate glyphs*. One possible benefit that a multivariate glyph may bring is the provision of an intermediate level of detail between a dot and its detailed attributes, which are displayed in a popup window activated by hovering over the device over the dot. Viewing a coarse depiction of file attributes may facilitate fast elimination of incorrectly returned search results, and reduction of the need for hovering over every dot (or glyph) to inspect its details.

DOI: 10.2312/cgvc.20141212



(a) (b) (c)

Figure 1: *The results of a search in a file system can be represented by a focus+context visualization, where the* context *is the file system shown as a treemap and* focus *is the collection of the files found. (a) The conventional depiction of search results as static colored-dots can present some difficulties in visual search. Animating these dots may make the focus standing out from the context. (b) Multivariate glyphs may be used to encode more file attributes than dots, but may introduce more occlusion of the context. (c) Cyclically-animated glyphs may facilitate visual search and attribute encoding, but may affect the perception of attributes or cause irritation.*

However, when using multivariate glyphs to depict search results as shown in Fig. 1(b), the focus information may sometimes cause serious occlusion of the context information. Naturally, this leads to design option of animating glyphs cyclically as shown in Fig. 1(c). Such glyphs are referred to as *cyclically animated glyphs* (*CAG*).

Our multi-faceted study thereby examined three alternatives to static dots, namely *(a)* cyclically-animated dots, *(b)* static glyphs, *(c)* cyclically-animated glyphs. Our theoretical rationales for considering cyclic animation and multivariate glyphs include:

- Animated motion, especially the onset of a motion, captures attention (e.g., [PRGA10, AC03]). Motion pop-out effects have been observed in infants as young as 4.5 months old [Dan05]. "Visual search for a moving target among stationary distractors is more efficient than searching for a stationary target among moving distractors" [Ros99]. All these suggest that motion vs. static can provide focus and context information with different levels of attention stimulation.
- The human visual system and memory system allow us to capture multiple pieces of information displayed in an alternating manner at the same location. It is widely accepted that information about a visual stimulus remains in the sensory memory for up to one second [Bad66], and in working memory between 15-30 seconds [PP59]. If attentively rehearsed and exercised (common in visualization), the retention of information within working memory can increase. If multiple data attributes are integrated into a visual representation (e.g., a multivariate glyph), then working memory may hold still more information [Mil56]. This suggests that it is feasible to use cyclic animation to address the problem of occlusion as viewers are able to perceive both focus and context information, while the cyclic nature allows the visualization to remain effective as a tool of external memorization.
- Although animation may not always be effective in visualization (e.g., [RFF*08,Fis10]), the primary use of cyclic animation considered in this study is not for conveying information, but rather for capturing attention, and resolving or reducing the problem of occlusion. Hence it is possible that cyclically animated dots and glyphs may not suffer from the conventional drawbacks associated with animation in visualization, such as lack of external memorization capability, inaccuracy in motion perception and motion comparison, and poor pop-out effects in motion vs. motion comparison.

However, these theoretical rationales may not guarantee that any of the three alternative approaches will offer effective visualization. In particular, glyph-based visualization inevitably adds an additional burden upon users as there is the need to learn and memorize glyph encoding in order to benefit from the use of multivariate glyphs. Therefore an empirical study was conducted in the form of a focus group comprising a short survey followed by an open discussion. The rationale for choosing this qualitative format is that it allows us to explore the possible merits and demerits of the three alternatives from different aspects, which would be difficult were a task-based quantitative experiment conducted.

The empirical study enabled us to gain several new insights. Firstly, all three alternatives can stimulate more attention in viewing search results overlaid on top of a treemap. Secondly, it is still feasible to perceive different attribute values even when a glyph is animated cyclically. Thirdly, participants preferred cyclically-animated dots and static glyphs, while most found cyclically-animated glyphs somehow irritating. Finally, participants expressed a fair level of willingness to learn and memorize glyph encoding, and many showed their capability of understanding a given encoding with little effort.

2. Related Work

Our work builds on research in the areas of treemaps, search results, and glyph-based visualization. We therefore briefly review the related work in these areas.

Treemaps. The concept of a treemap was proposed by Shneiderman [JS91], initially for visualizing the space utilization of a hard drive. Since then, the use of treemaps has been extended to almost any data with a hierarchical organization. Treemaps and pixel-based visualization share a similar dense appearance and pixel-based visualizations can also be regarded as a special case of a treemap. Keim provided a comprehensive survey of pixel-based visualization [Kei00]. Oelke *et al.* proposed a number of methods for boosting some focus information in a pixel-based visualization [OJS*11].

Glyphs. Glyphs are visual entities composed of several visual channels representing multivariate attributes. Glyph designs tend to be largely influenced by particular application domains and tasks. Here, we mainly review the glyph-based visualization techniques relevant to information visualization. Ward [War02] classified different glyph placing strategies, proposing rules for their usage and noted several bias issues in the context of information visualization. He proposed a perceptually-aware process of glyph generation via the mapping of data attributes to visual channels in [WO08]. Yang *et al.* meanwhile proposed Value and Relation (VaR) for visualizing multidimensional large data sets [YHW*07]. In information visualization, ring-shaped glyphs and radial mapping are common [HLNW11]. Further techniques on radial methods for information visualization were surveyed by Darper *et al.* [DLR09]. A comprehensive review of glyph-based visualization for spatial multivariate medical data is proposed in [ROP11]. Glyph-based visualization has also been applied in a wide range of domain specific data and tasks such as medicine [MSZB10, KW06, ZPB09], software [CE98], text [RSE98], literature [ARLC*13, Che06], computation [WPL96], scientific visualization [HLNW11, FH09], geovisualization [KJC*09, FS04, Ber83], sports [LCP*12], sensor data [BMJK09, ZFH08], and biology [MRSS*12, MRS*09].

Search Results Visualization. Here we present focus+context and glyph-based techniques applied to information search result visualization. xFind [ASL*01] was developed to display search results as a ranked list as well as a scatter plot. Roberts *et al.* used glyph to visualize web search results [RBR02]. Hoeber and Xue [HY06] developed HotMap and Concept Highlighter to show search results as two levels of detail: an overview of the top 100 documents, and a detailed view of 20–25 documents at a time. They also discussed how these views support the visual query of web search results. Chau [Cha11] proposed a glyph based on a flower metaphor to visualize web search results.

Animation in Visualization. The effectiveness of animation in visualization has been a debatable subject. In general, most agree that animation is an effective tool to aid learning and story telling (e.g., for explaining algorithms [TMBH02], or volume visualization [WH07]). Pritzkau *et al.* used animation for the visual boosting of scatter plots [PB10]. Farrugia and Quigley [FQ11] used animation for visualizing temporal graphs. Robertson applied animation for presenting trends in multivariate data [RFF*08]. Ware and Bobrow [WB06] used animation in motion coding for patterns.

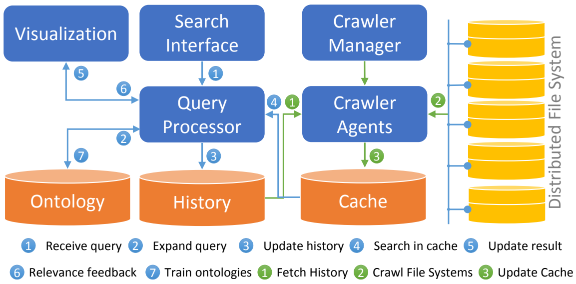


Figure 2: *The system architecture as used in our case study.*

3. Scenario: Search in File Systems

The capacities of file systems are increasing rapidly, along with the growing activities that rely on data sharing and data repositories. Searching files, especially among unfamiliar volumes of files created by others (e.g., collaboration partners, subcontractors, etc.), is not a trivial task. Over the past decade, many operating systems have been equipped with advanced search indexing facilities (such as Spotlight on OS X and Zeitgeist or Catfish on Ubuntu GNU/Linux) to alleviate some of the difficulties of search files. However, file searching is a complex cognitive process, and its performance depends on numerous factors, many of which reside with the human users. Hence, no search technique can be guaranteed to find all files when the users are unsure of exactly what files they are looking for and how to correctly specify the search criteria. A file search is often a explorative process during which the users often progressively refine the search criteria by observing and learning from errors in the earlier search results. However, operating systems typically present search results in a list or tabular view, which is usually accompanied by a sorting mechanism. One shortcoming of a list or tabular view is the lack of contextual information. It is often difficult to have an overview about how search results are distributed across different parts of the file system, and difficult to build a mental model about the storage structure around each search results, the clustering patterns among search results, or the relationships between different search result. Without such contextual information, it is difficult to judge whether a search is successful or not, or how to refine search specification.

Treemaps were introduced to visualize file system data [JS91]. They provide a compact display of directory hierarchy, allowing additional visual encoding of one or two file attributes such as size and type. It can offer a cost-effective means of displaying the contextual information for a set of search results.

In this work, we consider situations that involve a relatively large file system, where files may be distributed across different locations on a network. The file system is shared by many users, and contains a large volume of files at a scale (e.g., petabyte or exabyte) well beyond that of an ordinary desktop computer. Many practical working environments feature such file systems nowadays. Some users, such as an adminstratctor of a large project team or a data archivist in an organization, have to perform such file system searches frequently. Because of the large number of files in such a file system, it is not feasible to depict every file in a treemap. We therefore expect a treemap to show only 3-5 levels of the hierachy rather than showing the complete recursion. Most treemap elements will be folders rather than files. Hence a file found by a search may not be mapped directly to an individual treemap element. Most likely, it is displayed as an object on a treemap element, which is interpreted as a "file inside this parent, grandparent or ancestor folder". It is relatively trivial to provide these objects with "details on demand", for instance, by opening a popup window when the pointer is hovering over an object. This in effect is a focus+context visualization, with search results as the focus, and the treemap representation of a file system as the context. As discussed in Section 1, the objects that represent search results may be displayed as static dots, cyclically - animated dots, static glyphs and cyclically-animated glyphs.

The development of such focus+context visualisation for file searches was motivated by the development of a search engine for very large file system. This search engine optimizes the speed of each search by building a cache of the files that are most relevant to each search specification. As illustrated in Fig. 2, the system has four main functional modules, namely *user interface*, *search history database*, *crawlers*, and *ontologies*. Users enter the search query through the user interface. The returned search result consists of metadata information of the files found.

The queries performed by the users are stored in the history database and used by the crawlers while building the cache. Each crawler is an agent that scans through the file system in an automated manner at predefined scheduled intervals, and fetches the information about the most relevant documents based on history database. The ontologies store various relationships among file attributes and terms. The details of this search engine are not the focus of this work. However, because this search engine relies on cached search results to a large extent, it is necessary to provide an effective means for users to observe the search results. Should the cached search results be unsatisfactory, the users can invoke a more detailed full-scale search, though given the size of the search across multiple disks across multiple servers, such a search could take several hours.

4. Visual Mapping of Glyphs

The glyphs in our system are composed of several visual channels presenting the multivariate attributes associated with the EXT4 file system data. In an EXT4 file system, the *stat* structure contains the *inode* information and stores 13 different attributes related to a file. Encoding these 13 attributes in a single glyph posed a significant challenge, mainly because *(a)* it requires a high pixel resolution to encode all attributes; *(b)* visual channels would lose their discriminating capacity under low-resolution conditions; *(c)* the visual context is dense, and *(d)* the time attribute (*time_t*) contains multiple components: time and calendar date; hence, two different visual channels are required to encode each time attribute.

We began our design activities with a series of brainstorming meetings. In these meetings, we identified the most commonly-used file attributes: type, size, time, date and protection mode. We produced a table consisting of several optional visual channels for each file attribute. After several iterations, a suitable option for each file attribute was identified on the basis of occlusion, clarity, visual separation, metaphoric abstraction for aiding learning and remembering, and encoding costs in terms of scalability. A schematic diagram of our final glyph design is shown in Fig. 3.

File type. Both color hue and shape are effective visual channels for categorical attributes. We considered shape as a visual channel for mapping the file types. However, there is a strong interdependence between shape and size. When the available display size is small, the number of discriminable variations for shapes becomes small. File types are thus represented as saturated colors on a five-pixel dot. We adapted the file categorization and color-coding scheme of GNU/Linux, and divided files into *n* categories, each is assigned to a light color (assuming a dark background). By default, we have 8 categories of files. Both the categorization and color mapping can be modified by users.

Size. The file size is a numerical attribute. We considered several design options, including the area of a shape, and length of an arch. However, a file system consists of diverse varieties of different sized files. The same visual channel cannot support discriminable variations at the gigabyte level as well as the kilobyte level. We thus divided a value of file size into four components, 0-999 bytes, 0-999 kB, 0-999 MB, 0-999 GB. Each component is shown as a pale yellow pie segment at one of the four quadrants of the glyph. This enables users to focus selectively on a specific quadrant, for instance, in searching for a specific range of sizes, or for comparing file sizes indicated by different glyphs. For file size of a TB or above, we replace each yellow segment with a red segment, such that TB in place of bytes, PB in place of KB, EB in place of MB and ZB in place of GB. The radius of the pie segment is 40 pixels and the spanning angle is 70°.

Time. The clock metaphor is a natural choice for encoding the time attribute. We explored several designs of the metaphor based on a pointer moving around a circular object. In order to cover the range of [00:00:00, 23:59:59], we used a spiral to facilitate the 720° rotation of the pointer. Because it is not intuitive to determine which half of the range

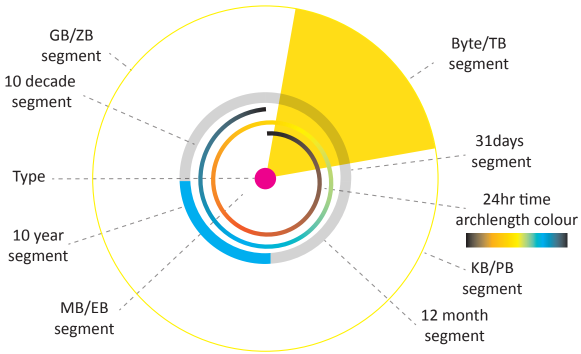


Figure 3: *File attribute glyph. Four different parameters (type, size, date and time) are encoded into the glyph.*

the inner (or outer) circular segment represents, we added a metaphoric color map. As shown is Fig. 3, the continuous colormap features black, orange, yellow and cyan at the 0, 6, 12, 18 hour positions, symbolizing night, morning, noon and afternoon respectively. The combination of spiral location and color symbolism provides a dual visual encoding of time for fault tolerance in perception. The largest and smallest diameter of the spiral is 38 and 30 pixels respectively.

Date. The date attribute in a file system is normally displayed in text using one of the international or regional formats (e.g., YYYY-MM-DD). In the context of file searching, it is more useful to encode the age of a file, i.e., the interval between the date of searching and the record date of creation (or last modified/accessed). The main challenge in visually encoding the date attribute is the large range of values, and different search tasks usually focus on a specific range. We thus adopted a design concept similar to that for **size**. We divided file ages into four categories: day, month, year and decade. A translucent grey ring of 40 pixels in diameter is divided in four quadrants. The first quadrant represents the range of 0-31 days, the second 0-12 months, the third 0-10 years, and the forth quadrant 0-10 decades. A non-zero value in each quadrant is encoded as the length (or degree) of a colored circular arc, extending clockwise. We considered the option to display creation, modification and access time/date all together simultaneously. However, we could not find a suitable mechanism to separate these six attributes visually. Therefore, we decided to show one attribute for date and one for time, with an option for users to choose different attributes.

Protection mode. The protection mode attribute is composed of nine access permissions for [read, write and execute] × [user, group and others]. We tried out a number of design options. They all resulted in noticeable visual clutter. We thus decided not to consider glyph encoding of the protection mode in this study, though we have implemented one design in our system as an option requiring explicit activation by the user.

In our user interface, glyphs are displayed on top of a squarified treemap [BHV99] showing the search space. The default search space is set according to the access status of a user, e.g., the entire file system for a superuser. The treemap displays the top k-levels of the search space. As we focus this study on large file systems, we assume that not all files or directories in the search space can be represented by the treemap. For example, our testing environment consists of more than 500,000 files. We thus use the colors dark green and purple to represent directories and files respectively. We indicate different directory levels through the use of different shades of green and purple, with lighter shades indicating higher levels, and darker shades indicating lower levels. The default depth k is set to 5.

Once the user enters their search criteria, the results of the search are shown as glyphs overlaid on top of the treemap, as shown in Fig. 4. This figure shows around 90 search results on top of a treemap with five levels. As this study also includes static dots, cyclically-animated dots and cyclically-animated glyphs, these are all options available through a *visual mapping dialog window*. The dialog window is used mainly to define the mapping of file attributes to visual channels of the glyphs. For cyclic animation, there are additional options for choosing the animation style and adjusting the animation tempo, which will be discussed in the next section.

Cyclic Animation. We considered several animation styles, namely changing size, color, and opacity. Because color is a highly attentive visual channel while the perception of color encoding is easily confused, we ruled out color-based animation at an earlier stage of the design. Both size and opacity based animation have their relative merits and demerits. In addition, whereas a single animation style can be applied to a glyph, one can also apply different animation tyles to different visual channels of a glyph.

5. Evaluation

5.1. Organization

After implementing a file search interface with four different design options, namely *(a)* static dots, *(b)* cyclically-animated dots, *(c)* static glyphs, and *(d)* cyclically-animated glyphs, We conducted a further empirical study to compare these four design options. Although such a study may theoretically be carried out in the form of controlled experiments, the range of factors that may affect the user performance is rather extensive. As there has not been any previous study on this topic, the first step of evaluation was in fact to establish a list of such factors. We thus chose one of the commonly-used qualitative research methodologies, *focus group discussions*. The participants were 10 individuals from the staff and postgraduate student communities. All had experience of using search facilities currently available in various file systems. In order to achieve a good balance between collective views from the group, and strong voices within the group, and be-

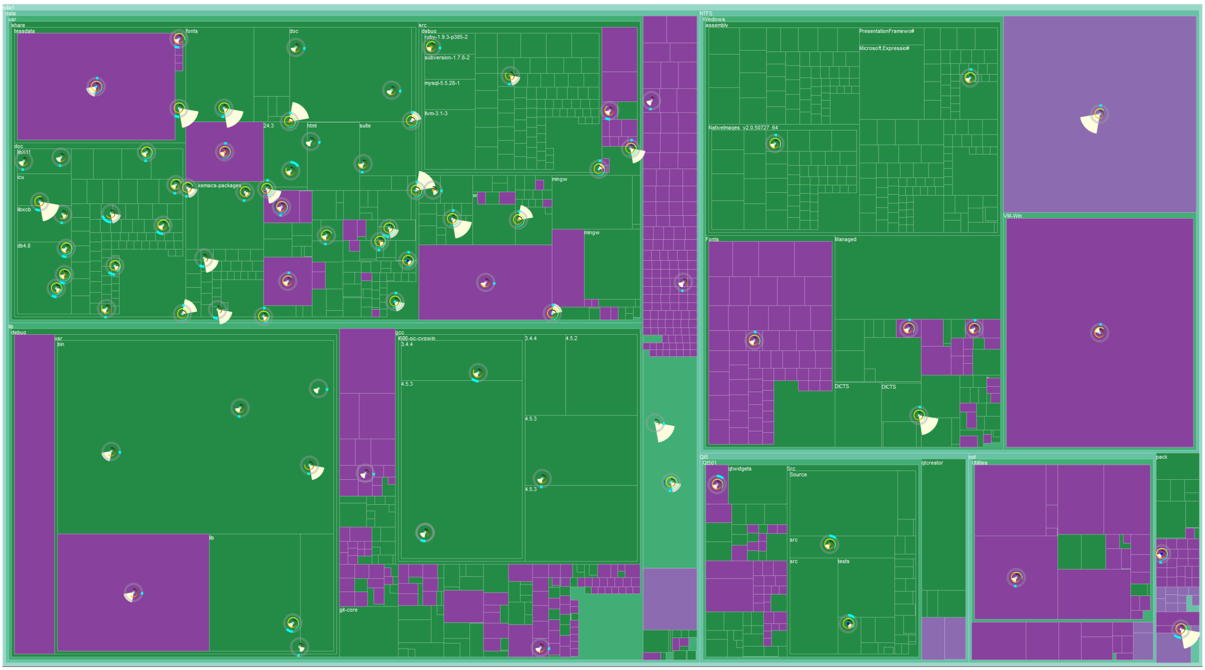


Figure 4: *Visualization of file search, featuring glyphs for search results (focus) and a treemap for the file system (context).*

tween learning visualization techniques and offering opinions, we organized the participants into two equal-size group meetings, both with the same schedule plan.

Introduction (20 minutes). Each meeting began with a 20 minute introductory presentation. Participants were informed of the overall goal of the system, application scenarios, and objectives of the meeting. They were then introduced to the concepts of treemap, glyph-based visualization, and four design options, each accompanied by a relatively comprehensive list of possible pros and cons compiled by the research team. The word "possible" was emphasized to avoid leading the participants. By giving a balanced list of possible pros and cons, participants were made to feel that they were being consulted rather being tested.

Survey (20 minutes). Following the introduction, participants were presented with a set of 14 statements and asked to rate, using a 5-point Likert scale, the extent to which they agreed or disagreed with each statement. This allowed us to collect views from every individual participant in an equal manner, and to compile them into a coherent set of collective views. The 14 questions, provided for reference in the supplementary materials, were divided into four groups, namely *(a)* static dots vs. animated dots, *(b)* static dots vs. static glyphs, *(c)* static glyphs vs. animated glyphs, and *(d)* animated dots vs. animated glyphs. When answering the questionnaire, example visualizations were shown side-by-side using two projectors.

Open Discussions (20 minutes). Following the questionnaire session was an open discussion session, during which participants were encouraged to offer feedback and comments relating to design issues, especially those were not covered by the questionnaire. In both meetings, the discussions were lively, and some participants were highly enthusiastic in offering their opinions and suggestions.

5.2. Survey Results

Static dots vs. animated dots. Results show a strong preference towards animated dots. 80% of participants expressing such a preference, only 10% of participants preferring static dots, and 10% expressing no clear preference. 40% of participants rated animated dots as either mildly annoying, or annoying. This indicated that cyclic animation can provide additional attentive power, which is necessary for identifying small objects distributed around a treemap.

Static dots vs. static glyphs. The results indicate a willingness to learn the visual encoding of the glyph (90%), even though many found such learning and remembering to be difficult (70% and 60% respectively). Many (60%) agreed that glyphs would be advantageous when one needs to perform 20 or more searches per day .

Static glyphs vs. animated glyphs. Results show a general preference for static glyphs. Most found it easier to identify individual search results (60%), and see individual file attributes (70%). Most rated animated glyphs as either mildly annoying or annoying (80%).

Animated dots vs. animated glyphs. As expected, there was little difference when asked which afforded better views of the underlying directory structure, with 50% and 40% rating animated dots and animated glyphs respectively as giving the best view.

5.3. Summary of Open Discussions

During the discussions, participants asked several questions on the visual encoding of the glyph and showed much interest to learn. Participants learned the visual encoding very quickly, and offered informed suggestions about various design aspects. For example, one pointed out that the visual encoding of the size attribute should support file sizes of a terabyte or more. We accommodated this suggestion into a revised design. Some participants pointed out an important factor that even without any animation a glyph is easier to see than a dot, due to the size factor. Naturally, there were some suggestions with regards to our use of colour, with a number of these leading to debate between participants.

The participants made a number of positive comments. In particular, such visualization can be useful in quickly interpreting the search results and their location in a file system instead of reading any detail text. Some expressed that in a traditional file system search, when the search results were displayed they had to go through each result one by one. Sometimes in order to locate a single result, users are required to navigate through several hierarchical levels. Repeatedly performing such tasks was a very tedious job. The use of this new visualisation could eliminate this problem, and could therefore prove to be very useful.

There was significant discussion regarding the use of treemaps as the background context, with some participants finding it difficult to appreciate the usefulness of a recursive directory-based cell layout verses their own alternative suggestions such as a calendar layout. These discussions also confirmed that the use of animated dots was useful, as static dots were, at times, difficult to see, whilst views on the animated glyphs remained negative. Many participants felt that even when animation frequency was slowed, animation could still prove distracting. Some participants appreciated the options provided in the user interface for switching on and off individual visual channels as reading and comparing a specific attribute is easier.

6. Conclusion

We conducted a study on glyph-based visualization over a dense visual context. We used a file search in a large system as the scenario to facilitate the exploration of various design options as well to provide a frame for the empirical study.

Our consultation of the potential users in a focus group study indicated the following: (a) Users are generally willing and able to learn glyph encodings if the task of file search needs to be performed frequently within a large file system. (b) Users generally preferred cyclically-animated dots and static glyphs as opposed to static dots and cyclically-animated glyphs. (c) Potential users are in agreement that static dots are ineffective in supporting visual searches due to their limited visibility. (d) Most users found cyclically animated glyphs somewhat irritating in a visual interface, though they can be a workable option for some.

As the visualization problem studied in this work exhibits a huge design space, this study has addressed only a small scope within such a design space. We suggest that future studies may obtain user performance data through task-driven experiments under more controlled settings. As such task performance may be related to several aspects, such as accuracy, speed, learning, memorization, and levels of irritation, it would likely require several empirical studies in order to obtain statistically-significant data.

7. Acknowledgement

The authors would like to thank Laing O'Rourke for sponsoring the DPhil studentship of the first author.

References

[AC03] ABRAMS R. A., CHRIST S. E.: Motion onset captures attention. *Psychological Science 14*, 5 (2003), 427–432. 2

[ARLC*13] ABDUL-RAHMAN A., LEIN J., COLES K., MAGUIRE E., MEYER M., WYNNE M., JOHNSON C. R., TREFETHEN A., CHEN M.: Rule-based Visual Mappings, with a Case Study on Poetry Visualization. *Computer Graphics Forum 32*, 3 (2013). 3

[ASL*01] ANDREWS K., SABOL V., LACKNER W., GÜTL C., MOSER J.: Search Result Visualisation with xFIND. In *Proc. 2nd International Workshop on User Interfaces to Data Intensive Systems* (2001), p. 50. 3

[Bad66] BADDELEY A. D.: Short-term memory for word sequences as a function of acoustic, semantic and formal similarity. *Quarterly Journal of Experimental Psychology 18*, 4 (1966), 362–365. 2

[BdHV94] BRIDGEMAN B., DER HEIJDEN A. H. C., VELICHKOVSKY B. M.: A theory of visual stability across saccadic eye movements. *Behavioral and Brain Sciences 17*, 02 (1994), 247–258. 1

[Ber83] BERTIN J.: *Semiology of Graphics*. University of Wisconsin Press, 1983. 3

[BHV99] BRULS M., HUIZING K., VAN WIJK J. J.: Squarified Treemaps. In *In Proc. Joint Eurographics and IEEE TCVG Symposium on Visualization* (1999), pp. 33–42. 5

[BMJK09] BAK P., MANSMANN F., JANETZKO H., KEIM D. A.: Spatiotemporal analysis of sensor logs using growth ring maps. *IEEE Transactions on Visualization & Computer Graphics 15*, 6 (2009), 913–920. 3

[CE98] CHUAH M. C., EICK S. G.: Information rich glyphs for software management data. *IEEE Computer Graphics & Applications 18*, 4 (1998), 24–29. 3

[Cha11] CHAU M.: Visualizing web search results using glyphs: Design and evaluation of a flower metaphor. *ACM Transactions on Management Information Systems 2*, 1 (2011), 1–27. 3

[Che06] CHEN C.: CiteSpace-II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology 57*, 3 (2006), 359–377. 3

[Dan05] DANNEMILLER J. L.: Motion Popout in Selective Visual Orienting at 4.5 But Not at 2 Months in Human Infants. *Infancy 8*, 3 (2005), 201–216. 2

[DLR09] DRAPER G. M., LIVNAT Y., RIESENFELD R. F.: A survey of radial methods for information visualization. *IEEE Transactions on Visualization & Computer Graphics 15*, 5 (2009), 759–776. 3

[FH09] FUCHS R., HAUSER H.: Visualization of Multi-Variate Scientific Data. *Computer Graphics Forum 28*, 6 (2009), 1670–1690. 3

[Fis10] FISHER D.: Animation for visualization: opportunities and drawbacks. In *Beautiful Visualization*, Steele J., Iliinsky N., (Eds.). O'Reilly Media, 2010, ch. 19, pp. 329–352. 2

[FQ11] FARRUGIA M., QUIGLEY A.: Effective temporal graph layout: A comparative study of animation versus static display methods. *Information Visualization 10*, 1 (2011), 47–64. 3

[FS04] FUCHS G., SCHUMANN H.: Visualizing abstract data on maps. In *Eighth International Conference on Information Visualisation* (2004), pp. 139–144. 3

[HLNW11] HLAWATSCH M., LEUBE P., NOWAK W., WEISKOPF D.: Flow radar glyphs–static visualization of unsteady flow with uncertainty. *IEEE Transactions on Visualization & Computer Graphics 17*, 12 (2011), 1949–1958. 3

[HY06] HOEBER O., YANG X. D.: Exploring Web Search Results Using Coordinated Views. In *International Conference on Coordinated and Multiple Views in Exploratory Visualization* (2006), pp. 3–13. 3

[JS91] JOHNSON B., SHNEIDERMAN B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *IEEE Conference on Visualization* (1991), pp. 284–291. 1, 3

[Kei00] KEIM D. A.: Designing Pixel-Oriented Visualization Techniques: Theory and Applications. *IEEE Transactions on Visualization & Computer Graphics 6*, 1 (2000), 59–78. 1, 3

[KJC*09] KARNICK P., JESCHKE S., CLINE D., RAZDAN A., WENTZ E., WONKA P.: A Shape Grammar for Developing Glyph-based Visualizations. *Computer Graphics Forum 28*, 8 (2009), 2176–2188. 3

[KW06] KINDLMANN G., WESTIN C.-F.: Diffusion Tensor Visualization with Glyph Packing. *IEEE Transactions on Visualization & Computer Graphics 12*, 5 (2006), 1329–1335. 3

[LCP*12] 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 (June 2012), 1255–1264. 3

[Mil56] MILLER G.: The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review 63*, 2 (1956), 81–97. 2

[MRS*09] MÜLLER H., REIHS R., SAUER S., ZATLOUKAL K., STREIT M., LEX A., SCHLEGL B., SCHMALSTIEG D.: Connecting Genes with Diseases. In *13th International Conference Information Visualisation* (2009), pp. 323–330. 3

[MRSS*12] 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 & Computer Graphics 18*, 12 (2012), 2603–2612. 3

[MSZB10] MÜLLER H., SAUER S., ZATLOUKAL K., BAUERNHOFER T.: Interactive Patient Records. In *14th International Conference Information Visualisation* (2010), pp. 252–257. 3

[OJS*11] OELKE D., JANETZKO H., SIMON S., NEUHAUS K., KEIM D. A.: Visual Boosting in Pixel-based Visualizations. *Computer Graphics Forum 30*, 3 (2011), 871–880. 3

[PB10] PRITZKAU A., BARTZ D.: Scattering and Jittering : Using Real and Illusionary Motion for Better Visual Scatterplot Analysis. In *IEEE InfoVis conference* (Salt Lake City, USA, 2010). 3

[PP59] PETERSON L. R., PETERSON M. J.: Short-term retention of individual verbal items. *Journal of Experimental Psychology 58*, 3 (1959), 193–198. 2

[PRGA10] PRATT J., RADULESCU P. V., GUO R. M., ABRAMS R. A.: It's Alive!: Animate Motion Captures Visual Attention. *Psychological Science 21*, 11 (2010), 1724–1730. 2

[RBR02] ROBERTS J., BOUKHELIFA N., RODGERS P.: Multi-form glyph based web search result visualization. In *6th International Conference on Information Visualisation* (2002), pp. 549–554. 3

[Ren02] RENSINK R. A.: Failure to see more than one change at a time. *Journal of Vision 2*, 7 (2002), 245. 1

[RFF*08] ROBERTSON G., FERNANDEZ R., FISHER D., LEE B., STASKO J.: Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization & Computer Graphics 14*, 6 (2008), 1325–1332. 2, 3

[ROP11] ROPINSKI T., OELTZE S., PREIM B.: Survey of glyph-based visualization techniques for spatial multivariate medical data. *Computers & Graphics 35*, 2 (2011), 392–401. 3

[Ros99] ROSENHOLTZ R.: A simple saliency model predicts a number of motion popout phenomena. *Vision Research 39*, 19 (1999), 3157–3163. 2

[RSE98] ROHRER R. M., SIBERT J. L., EBERT D. S.: The Shape of Shakespeare: Visualizing Text using Implicit Surfaces. In *IEEE Symposium on Information Visualization* (1998), pp. 121–129. 3

[SR05] SIMONS D. J., RENSINK R. A.: Change blindness: Past, present, and future. *Trends in cognitive sciences 9*, 1 (2005), 16–20. 1

[TMBH02] TVERSKY B., MORRISON J. B., BETRANCOURT M., HALL J.: Animation: can it facilitate? *International Journal of Human-Computer Studies 57*, 4 (2002), 247–262. 3

[War02] WARD M. O.: A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization 1*, 3-4 (Dec. 2002), 194–210. 3

[WB06] WARE C., BOBROW R.: Motion coding for pattern detection. In *Proc. 3rd symposium on Applied perception in graphics and visualization* (2006), pp. 107–110. 3

[WH07] WOHLFART M., HAUSER H.: Story telling for presentation in volume visualization. In *Proc. 9th Joint Eurographics / IEEE VGTC conference on Visualization* (2007), pp. 91–98. 3

[WO08] WARD M. O., O. WARD M.: Multivariate data glyphs: Principles and practice. In *Handbook of Data Visualization*. Springer, 2008, pp. 179–198. 3

[WPL96] WITTENBRINK C. M., PANG A. T., LODHA S. K.: Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization & Computer Graphics 2*, 3 (1996), 266–279. 3

[YHW*07] YANG J., HUBBALL D., WARD M. O., RUNDENSTEINER E. A., RIBARSKY W.: Value and relation display: interactive visual exploration of large data sets with hundreds of dimensions. *IEEE Transactions on Visualization and Computer Graphics 13*, 3 (Jan. 2007), 494–507. 3

[ZFH08] ZHAO J., FORER P., HARVEY A. S.: Activities, ringmaps and geovisualization of large human movement fields. *Information Visualization 7*, 3-4 (2008), 198–209. 3

[ZPB09] ZHANG Y., PASSMORE P. J., BAYFORD R. H.: Visualization of multidimensional and multimodal tomographic medical imaging data, a case study. *Philosophical Transactions. Series A, Mathematical, physical, and engineering sciences 367*, 1900 (Aug. 2009), 3121–3148. 3