The Many-Faced Plot: Strategy for Automatic Glyph Generation

João Miguel Cunha, Evgheni Polisciuc, Pedro Martins and Penousal Machado
CISUC, Department of Informatics Engineering
University of Coimbra
Coimbra, Portugal
{*jmacunha,evgheni,pjmm,machado*}*@dei.uc.pt*

***Abstract*—Despite some authors stating that data-relatedness helps interpretation, glyphs are often used unrelated to the represented data. In order to automatically produce data-related glyphs, a large visual repository is required, as well as, image structure suitable for data representation. In this paper, we propose a strategy that fulfills the two requirements and allows the production of glyphs related to the data thematic (literal and metaphorical). We compare used approach with current glyph techniques and discuss the results.**

***Keywords*-Data Glyphs; Chernoff Faces; Information Visualization; Emoji;**

I. Introduction

Data glyphs are used in data visualization for the representation of multidimensional data [1]. Glyphs can be described as graphical objects that possess visual features, which can be assigned to data variables to produce a visualization.

Some glyphs are abstract (e.g. polygon glyph [2]) and others have an iconic nature (e.g. faces, cars, or even flowers). When considering iconic glyphs, they can be unrelated to the data thematic (e.g. a face glyph representing forest fires data) or be related in a literal (e.g. a face glyph representing data on facial features) or a metaphoric way (e.g. a face glyph representing non-facial anthropometric data).

Using glyphs related to the data is said to have perceptual advantages, which leads to easier interpretation and better accuracy [3]. In addition, some authors justify the usage of certain glyph designs with reasons related to human aptitudes, such as the ability to easily recognize faces, e.g. face glyphs [1], or to visually discriminate natural shapes, e.g. leaf glyphs [4]. For these reasons, several authors not only argue in favor of data-related glyphs, but also point out that it would be advantageous for a glyph-based visualization tool to have different types of glyphs, which could be chosen by the user and allow a better match to the data (e.g. [3]).

Such tool is normally considered difficult to implement as it would require a large repository of glyphs prepared for data-representation. Considering repositories such as image banks falls short, as images, due to their pixel-based nature, are often unsuitable for variation according to data.

On the other hand, we believe that emoji have several properties which make them adequate for this task. Emoji are

Figure 1: Examples of glyph variation for *face* thematic

pictorial symbols which were first made available in 1999 on Japanese mobile phones and in 2007 added to Unicode Standard[1], which led to their large scale usage. Firstly, it is beyond question emoji integration in written language, which is observed in the growing number of emoji-related tools and features – e.g. search-by-emoji supported by Google[2], and the Emoji Replacement and Prediction features implemented in iOS 10[3]. They have been applied in tasks such as sentiment analysis or word sense disambiguation (e.g. [5], [6]) and it is our belief that they can also be used in information visualization. Secondly, Twitter's Twemoji[4] consists of fully scalable vector graphics, which are suitable for glyph variation.

The glyphs, in particular Chernoff faces alike representations, are the core of the proposed work. The traditional process of glyph design is replicated and pushed towards an automatic way of glyph generation. To the best of our knowledge, there is no implemented system that allows glyph suggestion and variation based on the thematic used (see Fig. 1). We take advantage of the properties of emoji system and present an approach to implement an automatic glyph generation system. Our ultimate goal is to make the glyph design process simple and effective, facilitating the communication of complex information.

II. Related Work

Taking into account the nature of current work, this section provides an overview of glyph-based representation, discusses the principal aspects of automatic visualization, and covers metaphoric mapping in graphic representation.

[1] unicode.org/reports/tr51, retr. 2018.
[2] forbes.com/sites/jaysondemers/2017/06/01/could-emoji-searches-and-emoji-seo-become-a-trend/, retr. 2018.
[3] macrumors.com/how-to/ios-10-messages-emoji/, retr. 2018.
[4] github.com/twitter/twemoji, retr. 2018

A. Glyphs

In the context of information visualization, data glyphs are composite graphical objects that use their visual and geometric attributes to encode multidimensional data by mapping each dimension of data point to the marks of a glyph [7], [8]. A well known example of simple in design, yet complex and efficient in application glyph, is Star Glyph [9], [10], [11]. Star glyph consists of a number of uniformly separated arrayed lines that correspond to the number of data dimensions, where the lengths of each ray encode the magnitude of the corresponding data value, with the endpoints connected to form a polygon.

There are different kinds of glyphs with varying designs and conceptual diversity. Also, a variety of surveys about data glyphs and their usage have been published in the recent years [12], [13]. Nonetheless, this section focuses mainly on well known and most researched data glyphs – Chernoff faces [1]. This type of glyphs encode multidimensional data using facial features such as the length of the nose, orientation of the eyebrows, the shape of a mouth, among others. One particular feature of Chernoff faces, which originated other alike glyph designs, is its resemblance with a human face. Although this kind of glyphs perform poorly in terms of response time, as well as the accuracy of glyph decoding, when comparing to other existing glyphs [14], the metaphoric projection and analogy with faces make Chernoff faces a powerful tool for conveying complex data.

B. Metaphor

Numerous visualization methods resort to metaphors and analogies to make complex information understandable (e.g. Furnas [15], and Havre et al. [16]). Risch pointed out that saying "that is like this" is to make a poorly understood phenomenon more comprehensible by relating it to a more familiar visuospatial one [17]. In other words, graphical metaphor is a structural mapping from a source domain to a target domain, which allows the representation of complex phenomena in familiar visual terms. Rish distinguishes between analogical graphics (e.g. geographical visualization is seen as analogical mapping, because the source and target domains are both spatial in nature) and metaphorical graphics. Metaphorical graphics present non-spatial, abstract concepts in spatial terms, where both domains have different semantics. As the author demonstrated, it is more natural for humans to think about abstract data spatially. This is due to our daily experience with real world. Rish focuses mainly on the Gentner's structural-mapping model [18], which is similar to semantic networks, where knowledge is expressed with nodes (objects/concepts) and links (predicates that express propositions). Finally, the author argues, referring to 3 studies, that the visual analogy and metaphor are key aspects of human cognition, and play key role in everyday communication.

A recent work of Fuchs et al. is an example of metaphorical representation of abstract data that was inspired by nature [4]. This visualization metaphor depicts multidimensional data with glyphs inspired by environmental metaphor – a leaf. Another example is already mentioned Chernoff faces, which use metaphoric mapping to represent data by different features of a cartoon face: shape of the face, size, position and shape of nose and mouth [1]. Similarly, we intend to implement a metaphoric mapping in our system, which takes into account semantics of data and emoji. Moreover, our system allows the projection of data to any object of the target domain (e.g. car, leaf, face, house, etc.).

C. Automatic Visualization

In 1990's and beginning of 2000's there was increasing interest in intelligent or "smart" graphics, usually referred to as automatic visualization [19], [20], [21]. This approach mostly consists of a rule-based mapping between graphical elements and data type accompanied with additional underlying mathematical statistics, which are used to summarize the data. Automatic visualization is efficient to get first impression on the data. However, the disadvantage of this approach is the fact that automatic visualization is just statistical projections on visualization artifacts. It is extremely difficult to extract any insight or high-level information from such projections.

One example of automatic visualization is AutoVis tool, developed by Graham and Leland [22]. Given the dataset, the system decides how to appropriately visualize it basing solely on visualization theory and without presupposing a predefined visualization model. The overall strategy for selecting proper graphical elements is based on the Grammar of Graphics introduced by Leland Wilkinson [23]. The system uses a prioritizing mechanism to select most "interesting" views to display, which is similar to Google's Page Rank. Finally, the visualization utilizes statistical methods to find relationships in the given data, and represents it using a graph model, accompanied with simple graphics (e.g. area charts, bar charts, box-plots, etc.) that summarize computed statistics.

Another example of automatic visualization is the work of Mackinlay et al. [24]. The tool, called Show Me, integrates a set of user interface commands that enable automatic generation of tables of views for multiple fields in dataset. Provided with the specified rows and columns the tool automatically selects Mark Type – graphical element (e.g. bar, line, shape, etc.) – and View Type – graphical methods to represent data (e.g. scatter plot, bar chart, etc.). The decision is made based on the predefined rules, which result from – "best ways of producing charts and graphs". Finally, each additional command can yield a particular type of view. Likewise, our system is intended to generate data glyphs automatically based on the query that a user performs, or based on the data, in particular meta-data, provided to the system.

III. Approach

Glyphs are often produced using specialized programs, which have parametrizable functions to draw geometric shapes. Some authors propose their own programs (e.g. Glyph Explorer for producing car glyphs [3]). In this paper, we propose an approach for a tool which can be used to generate ready-to-use glyphs for data visualization. In order to achieve this, we take advantage of the emoji connection between pictorial representation and associated semantic knowledge. Our system uses the vector emoji images from Twitter – which are freely available – in combination with semantic knowledge gathered by Wijeratne [6].

Our end goal is to develop a semi-automatic system in which three general steps occur: (i) the user introduces the thematic; (ii) the system presents the user with possible glyphs for the introduced thematic, their visual variables and suggested variation ranges; (iii) the user selects the glyph and configures the assignment of data variables to visual variables, according to his/her preferences, and/or the data semantics.

A. Resources Used

This work builds upon the system behind the platform *Emojinating* [25] , which automatically retrieves emoji that represent concepts previously introduced by the user. It combines semantic network exploration with visual blending and, not only searches for existing emoji, but also produces novel ones. Emojinating's system has three components: (i) *Concept Extender* (uses ConceptNet to search for related concepts to a given one), (ii) *Emoji Searcher* (searches for existing emoji semantically related to a given word), and (iii) *Emoji Blender* (produces new emoji using visual blending). In the context of this project, we use the *Emoji Searcher (ES)* and the *Concept Extender (CE)*. The system integrates data from the following online open resources:

- Twitter's Twemoji 2.3: a dataset of emoji fully scalable vector graphics with 2661 emoji;
- EmojiNet: a machine readable sense inventory for emoji built through the aggregation of emoji explanations from multiple sources [26]. It provides semantic knowledge to the emoji of Twemoji dataset. The version used has data regarding 2389 emoji;
- ConceptNet: a semantic network originated from the project Open Mind Common Sense [27], used to obtain concepts related to a given one.

The ES takes words as input and retrieves emoji that are semantically related to them, using semantic data from EmojiNet regarding *emoji name*, *emoji definition*, *keywords*, *associated senses* and *unicode*. The results obtained can be directly related to the introduced word – emoji that matches the word – or indirectly related – emoji that matches related words retrieved using CE. This allows the system to supply the user with emoji that can be used as glyphs for an introduced thematic.

B. System Architecture

The system uses the following 4-step pipeline:

1) Identifying the topic: This step consists in the identification of the topic to be searched. This user-provided information is used to gather emoji to be glyph candidates. Depending on the type of data, the introduced topic may be a thematic (e.g. functioning *mode a*, described in subsection III-C) or categories of the data itself (e.g. functioning modes *b* and *c*).

2) Glyph generation: after gathering candidates to be glyphs, these are filtered (removing inadequate ones) and the remaining ones are prepared to be used in the visualization. This step can be divided into the following tasks:

- Emoji deconstruction: dividing the emoji into individual shapes. This is possible due to the layered-structure of the emoji dataset;
- Identifying visual variables: analyzing the individual shapes and assessing their quality as visual variables. Their quality may depend on their size and potential to be transformed (e.g. a small shape which is very close to another one is considered to have bad quality as, for example, its variation in size would lead to overlapping, and color variation would be low in salience due to its small size);
- Removal of inadequate emoji: emoji with number of possible visual variables lower than the number of data variables that need to be represented are removed;
- Evaluating visual variables: analyzing possible variations and providing the user with suggestions for the best variables to be used, based on how much variation is possible (e.g. calculating minimum and maximum sizes) and on effect on global glyph (salience).

3) The user configuration: the system presents the user with glyphs, their possible visual variables and suggested variation limits. Then user is able to configure the assignment of visual variables, as well as, establish the variation limits, with the help of the suggestions of the system;

4) Setup of the display: configuring the final view of the visualization, i.e. how the glyphs should be organized. Two examples of this are a grid layout or positioning according to a given parameter (e.g. positioning glyphs in a map according to geographic locations).

C. Functioning Modes

The system can be used in different ways, depending on data type, more specifically on the semantics of given data. We considered three functioning modes (see Fig. 2):

a) Glyph as single emoji: the emoji is used as a glyph and the individual shapes, in which it can be decomposed, are seen as visual variables. In such cases, the user obtains

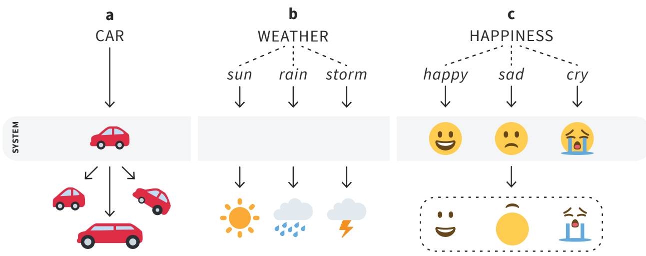

Figure 2: The three different functioning modes of the system

possibilities of glyphs for the introduced thematic (e.g. *car* thematic on the left side of Fig. 2). Similar to Chernoff faces, in this mode the applied data variables can be both categorical or numerical;

b) Glyph as set of emoji: the user does not introduce a single thematic but several categories. Emoji are retrieved for each category and they are used as a set. One example of is a weather map, in which each type of weather is represented by a different icon (e.g. sunny, rainy, etc.). This mode is intended to be related with categorical data;

c) Glyph as combination of emoji parts: similar to *mode b* but with the difference that the emoji for each category are merged into a single glyph, and the visual variables are not used in a continuous way, but as combinatory. In such cases, there is a high amount of shared features among the emoji for each category. The parts that change are identified and separated (e.g. in Fig. 2 for the *happy* category only the mouth is different from all the other emoji).

In all three modes, the user introduces a topic and receives emoji which represent it. These emoji are then transformed (mode a), used by replacement (mode b) or combined (mode c). The modes are related to the type of data: *mode a* will be used for the continuous data, whereas for situations in which there is only categorical data *modes b* or *c* will most likely be preferred. In *mode a* categorical visual variables may consist, for example, in color variation (e.g. the color of a car glyph).

IV. Discussion

This section provides critical discussion of the possible outcomes of our system. The analysis is divided into different sections: (i) comparison with existing glyph types (faces, cars and flowers), (ii) results for dataset thematics used by other authors, and (iii) open issues.

In order to analyze the potential of our approach in terms of usefulness in information visualization, we compared it with current glyph techniques. To do this, we conducted a bibliographic research which used the systematic review on glyphs by Fuchs [12] as a start point (later extended to other papers). We focused our search on iconic glyphs and collected a total of 40 research papers. These were analyzed and 13 were discarded as we considered that the glyph used was too abstract. For the final selection we collected the following information: type of glyph(s) used (e.g. car), number of total glyph visual variables, number of visual variables used, thematic of the dataset used and data variables represented. In this paper, we chose to only present some examples due to lack of space and also due to content overlap in some of the collected papers.

In current approaches, glyphs are either used of-the-shelf or custom made for the visualization at hand. In comparison, our system provides a multipurpose way of generating glyphs, without previously defined thematics.

On a general level, this approach allows the user to get emoji that represent the introduced topic. The system provides several emoji options to the user which, due to their layered-structure, are easy to prepare for a visualization task. The end version of the system is considered semi-automatic as the user is responsible for configuring the visualization.

The approach described in this paper is already partly implemented. Currently, the system retrieves emoji for introduced words (step 1 and part of step 2). As such, the glyphs presented in these paper were automatically gathered using the system.

A. Comparison with existing glyphs

The first analysis that we consider important to be performed has to do with the generation of glyph types that have been often used in information visualization. In our bibliographic research focused on iconic glyphs, we were able to identify three glyph types which we considered as benchmarks: face, car and flower. We used our system to obtain emoji that were similar to these glyph types and compared them (see Fig. 3).

The analysis of the glyphs produced by our system was focused on overall visual representation, the number of visual variables, and easiness of variation. The number of possible visual variables is more or less easy to calculate by decomposing the emoji into its individual shapes. Despite that, in some cases, the shape can be partially hidden or too small to apply transformations in a perceptible way and without influencing other shapes. Each shape can vary in terms of location, size, orientation, colour (hue,value and saturation), texture, among others. Obviously some shapes may be more suitable for some of the transformations – e.g. a color or texture variation in a very small shape will mostly likely have a very reduced effect on the perception of change. We estimated the number of visual variables for each of the glyphs obtained. The comparison with existing glyphs addressed two topics: visual appearance of the glyph and number of visual variables.

1) Faces: There are different versions of the face glyph, which vary in terms of number of visual variables (from 4 [30] to 32 [28]) and in terms of realism. Our system is able to produce different candidates to be used as face glyph, which visual differ among each other (e.g. not all

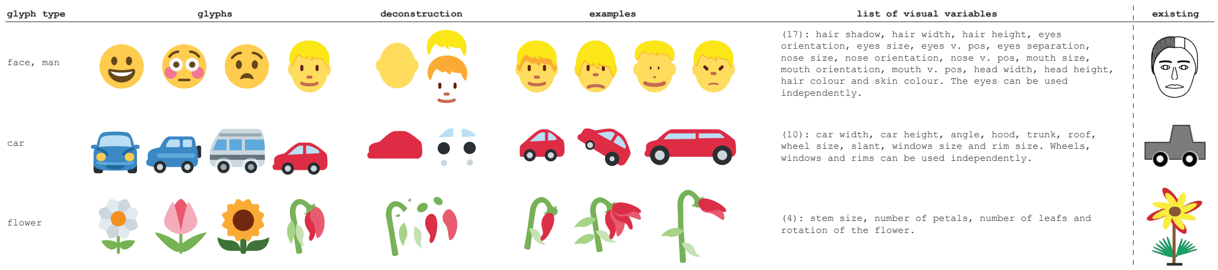

Figure 3: Examples of glyphs obtained with our system, their deconstruction, usage examples, list of possible visual variables and comparison with glyphs used by other authors (face [28], car [3] and flower [29]).

have hair or the inner shape of the eye). We are able to produce a face glyph with 17 visual variables. Despite this value being very distant from the one from Flury-Riedwyl face glyph (32 [28]), in the majority of the analyzed papers which employ face glyphs less than 15 visual variables were necessary to represent the data.

2) Cars: Concerning the car glyph, we are able to produce emoji with as many visual variables (10) as the ones from existing car glyphs [3]. We are also able to present the user with different versions (i.e. different types of car) which have higher number of visual variables. When comparing the appearance of our versions with the one from Siirtola [3], ours can be considered more realistic.

3) Flowers: Two versions different of flower glyph currently exist: one with stem [31], [29] (3-4 visual variables) and one with just a flower shape [2] (3 visual variables). Both implementations use functions which automatically produce any number of petals. As our approach uses emoji previously drawn, such visual variation would require a custom implementation. Despite that, we are able to match the number of visual variables of existing flower glyphs.

Overall, we are able to obtain similar number of visual variables to the ones from existing glyphs and we consider our versions more visually appealing, although further studies have to be made. Moreover, whereas implementation using functions allows greater flexibility, our approach allows higher variability of glyphs.

B. Thematic representation

In our bibliographic research we collected the thematics of all the datasets represented by the analyzed papers (see Fig.4) and used our system to produce glyph candidates for them. This allowed us to assess the performance of the system in suggesting data-related glyph candidates.

The majority of the analyzed papers used unrelated or metaphoric glyphs to represent the data (e.g. representing fires with a leaf [4]). With our system we were able to obtain data-related candidates for all the thematics, ranging from literal (e.g. a fire icon to represent fires) to metaphoric (e.g. a magnifying glass to represent Google search results).

The system is able to generate several possible glyphs for the same thematic. For example, for the thematic *fire*, we obtained 4 different candidates – going from literal (flame) to metaphorical (fire truck) – with different number of possible visual variables.

C. Overall Analysis and Open Issues

The glyph candidates obtained with the system are very different visually and in terms of structure (e.g. face has specific locations for the eyes, mouth and nose; a flower has a different behavior related to the positioning of petals). This leads to some issues that require further analysis.

1) Transformation: This structure difference affects the way transformations are applied. As already mentioned, whereas current glyph techniques use functions in glyph construction, we use previously designed emoji and the transformations are mainly done using distortion of existing shapes. As such, simple transformations like scaling and translation are easy to apply but complex transformations like duplication of shapes (e.g. flower petals) require custom implementation.

In addition, some shapes may be more suitable for some of the transformations than others. These aspects should ideally be analyzed for each glyph candidate and should be taken into account in order to define transformation rules to automatically suggest variation types and ranges.

2) Variation limits: our goal is to make the system automatically assess and suggest suitable ranges for each visual variable. To estimate these limits it is needed to calculate when shapes touch and overlap each other, following an approach similar to the one used in Cunha et al. [32].

3) Salience and Complexity: Some glyph characteristics will be difficult to assess due to the high variety of results. One example is the salience difference among visual variables of a glyph (some elements are more important than others in perception [38]). Another issue has to do with the complexity degree: some glyphs are very simple and others are much more complex (e.g. in Fig. 4 *fire* is simple and *education* is complex). This makes it necessary to further study the impact of glyph complexity on perception and interpretation and maybe limit the system to simpler emoji. The analysis of these characteristics would be impossible to do for every glyph and, as such, different approaches

glyph	word	thematic	glyph	word	thematic
	fire	Forest Fire [Fuchs 2015]		runner	Marathon runners [MacGregor & Slovic 1986]
	education	Investment in education [Li et al. 2015]		search	Google search results [Chau 2011]
	kill	Crime [Nelson et al. 1997]		student	Investment in education [Li et al. 2015]
	music	Classical Music [Chan et al. 2010]		temperature	Temperature / pressure [Casey 1987]
	note	Swiss-banknote [Hamner et al. 1987]		toxic	Toxic substances [Brown 1985]
	nut/seed	Seeds [Fuchs 2015]		twin	Twins [Flury & Riedwyl 1981]
	patient	Psychiatric patients [Mezzich & Worthington 1978]			

Figure 4: Glyphs obtained for dataset thematics used by other authors ([4], [2], [30], [33], [34], [35], [29], [36], [37], [28]).

should be followed (e.g. assessing salience using pixel-based difference calculation and limit the system to simpler emoji to avoid complexity).

Despite these open issues, in our opinion the major advantage of our approach is the automatic proposal of data-related glyphs, which we believe our system achieves. It is also important to mention that the emoji dataset can be easily changed, leading to different glyphs.

V. Conclusion

In this paper, we propose an approach for automatically generate data-related glyphs. Our system uses Twitter's Twemoji, which is suitable for data visualization due to the high conceptual coverage of Emoji system and its layered-structure image format. We assess the performance of our system by comparing generated glyphs with existing ones and its usefulness in representing datasets used by other authors. Future work includes: (i) implementing the remaining system components and (ii) testing it in real applications.

Acknowledgment

This project has been supported by Fundação para a Ciência e Tecnologia (FCT), Portugal, under the grants SFRH/BD/120905/2016 and SFRH/BD/109745/2015.

References

[1] H. Chernoff, "The use of faces to represent points in k-dimensional space graphically," *Journal of the American Statistical Association*, vol. 68, no. 342, pp. 361–368, 1973.

[2] Y.-N. Li, D.-J. Li, and K. Zhang, "Metaphoric transfer effect in information visualization using glyphs," in *Proc. of the 8th Int. Symposium on Visual Information Communication and Interaction*. ACM, 2015, pp. 121–130.

[3] H. Siirtola, "The effect of data-relatedness in interactive glyphs," in *Proc. of the Ninth Int. Conf. on Information Visualisation*. IEEE, 2005, pp. 869–876.

[4] J. Fuchs, N. Weiler, and T. Schreck, "Leaf glyph visualizing multi-dimensional data with environmental cues," 2015.

[5] B. Eisner, T. Rocktäschel, I. Augenstein, M. Bosnjak, and S. Riedel, "emoji2vec: Learning emoji representations from their description," in *Proc. of The Fourth Int. Workshop on Natural Language Processing for Social Media*, 2016.

[6] S. Wijeratne, L. Balasuriya, A. Sheth, and D. Doran, "Emojinet: Building a machine readable sense inventory for emoji," in *8th Int. Conf. on Social Informatics*. Springer, 2016.

[7] E. Anderson, "A semigraphical method for the analysis of complex problems," *Proc. of the National Academy of Sciences*, vol. 43, no. 10, pp. 923–927, 1957.

[8] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha, "Glyphs for visualizing uncertainty in vector fields," *IEEE Trans. on Visualization and Computer Graphics*, vol. 2, no. 3, 1996.

[9] J. H. Siegel, E. J. Farrell, R. M. Goldwyn, and H. P. Friedman, "The surgical implications of physiologic patterns in myocardial infarction shock," *Surgery*, vol. 72, no. 1, 1972.

[10] W. Peng, M. O. Ward, and E. A. Rundensteiner, "Clutter reduction in multi-dimensional data visualization using dimension reordering," in *Information Visualization, IEEE Symposium on*. IEEE, 2004.

[11] J. Yang, W. Peng, M. O. Ward, and E. A. Rundensteiner, "Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets," in *Information Visualization, Symposium on*, 2003, pp. 105–112.

[12] J. Fuchs, P. Isenberg, A. Bezerianos, and D. Keim, "A systematic review of experimental studies on data glyphs," *IEEE Trans. on Visualization and Computer Graphics*, vol. 23, no. 7, 2017.

[13] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, "Glyph-based visualization: Foundations, design guidelines, techniques and applications." in *Eurographics (STARs)*, 2013, pp. 39–63.

[14] M. D. Lee, R. E. Reilly, and M. E. Butavicius, "An empirical evaluation of chernoff faces, star glyphs, and spatial visualizations for binary data," in *Proc. of the Asia-Pacific symposium on Information visualisation-Volume 24*. Australian Computer Society, Inc., 2003.

[15] G. W. Furnas, *Generalized fisheye views*. ACM, 1986, vol. 17, no. 4.

[16] S. Havre, E. Hetzler, P. Whitney, and L. Nowell, "Themeriver: Visualizing thematic changes in large document collections," *IEEE Trans. on visualization and computer graphics*, vol. 8, no. 1, pp. 9–20, 2002.

[17] J. S. Risch, "On the role of metaphor in information visualization," *arXiv preprint arXiv:0809.0884*, 2008.

[18] D. Gentner, "Structure-mapping: A theoretical framework for analogy," *Cognitive science*, vol. 7, no. 2, pp. 155–170, 1983.

[19] E. D. Petajan, Y. D. Jean, D. Lieuwen, and V. Anupam, "Dataspace: An automated visualization system for large databases," in *Visual Data Exploration and Analysis IV*, vol. 3017. Int. Society for Optics and Photonics, 1997, pp. 89–99.

[20] S. M. Casner, “Task-analytic approach to the automated design of graphic presentations,” *ACM Trans. on Graphics (ToG)*, vol. 10, no. 2, pp. 111–151, 1991.

[21] Y. Iizuka, H. Shiohara, T. Iizuka, and S. Isobe, “Automatic visualization method for visual data mining,” in *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*. Springer, 1998.

[22] G. Wills and L. Wilkinson, “Autovis: automatic visualization,” *Information Visualization*, vol. 9, no. 1, pp. 47–69, 2010.

[23] L. Wilkinson, *The grammar of graphics*. Springer Science & Business Media, 2006.

[24] J. Mackinlay, P. Hanrahan, and C. Stolte, “Show me: Automatic presentation for visual analysis,” *IEEE Trans. on visualization and computer graphics*, vol. 13, no. 6, 2007.

[25] J. M. Cunha, P. Martins, and P. Machado, “How shell and horn make a unicorn: Experimenting with visual blending in emoji,” in *Proc. of the Ninth Int. Conf. on Computational Creativity (ICCC 2018)*, to appear.

[26] S. Wijeratne, L. Balasuriya, A. Sheth, and D. Doran, “Emojinet: An open service and api for emoji sense discovery,” in *11th Int. AAAI Conf. on Web and Social Media*, 2017.

[27] R. Speer and C. Havasi, “Representing general relational knowledge in conceptnet 5.” in *LREC*, 2012, pp. 3679–3686.

[28] B. Flury and H. Riedwyl, “Graphical representation of multivariate data by means of asymmetrical faces,” *Journal of the American Statistical Association*, vol. 76, no. 376, 1981.

[29] M. Chau, “Visualizing web search results using glyphs: Design and evaluation of a flower metaphor,” *ACM Trans. on Management Information Systems*, vol. 2, no. 1, 2011.

[30] E. Nelson, D. Dow, C. Lukinbeal, and R. Farley, “Visual search processes and the multivariate point symbol,” *Cartographica: The Int. Journal for Geographic Information and Geovisualization*, vol. 34, no. 4, pp. 19–33, 1997.

[31] B. Zhu, “Information visualization for knowledge repositories: applications and impacts,” 2002.

[32] J. M. Cunha, J. Gonçalves, P. Martins, P. Machado, and A. Cardoso, “A pig, an angel and a cactus walk into a blender: A descriptive approach to visual blending,” in *Proc. of the Eighth Int. Conf. on Computational Creativity*, 2017.

[33] W.-Y. Chan, H. Qu, and W.-H. Mak, “Visualizing the semantic structure in classical music works,” *IEEE Trans. on visualization and computer graphics*, vol. 16, no. 1, 2010.

[34] C. Hamner, D. Turner, and D. Young, “Comparisons of several graphical methods for representing multivariate data,” *Computers & Mathematics with Applications*, vol. 13, no. 7, 1987.

[35] D. MacGregor and P. Slovic, “Graphic representation of judgmental information,” *Human-Computer Interaction*, vol. 2, no. 3, 1986.

[36] E. J. Casey, “Visual display representation of multidimensional systems: The effect of system structure: And display integrality,” in *Proc. of the Human Factors Society Annual Meeting*, vol. 31, no. 1. SAGE Publications, 1987.

[37] R. Brown, “Methods for the graphic representation of systems simulated data,” *Ergonomics*, vol. 28, no. 10, 1985.

[38] G. De Soete, “A perceptual study of the fluryriedwyl faces for graphically displaying multivariate data,” *Int. journal of man-machine studies*, vol. 25, no. 5, pp. 549–555, 1986.