

Can the Grammar of Graphics Describe Data Glyphs? Exploring the Application of the Grammar

Barbara Nascimento*
Institute for Interdisciplinary
Research, University of
Coimbra

João M. Cunha†
CISUC/LASI – Centre for
Informatics and Systems of
the University of Coimbra

Evgheni Polisciuc‡
CISUC/LASI – Centre for
Informatics and Systems of
the University of Coimbra

ABSTRACT

The Grammar of Graphics (GoG) provides a formal and compositional framework for describing how data are transformed into graphical representations. Although grammar-based approaches have been widely applied to conventional visualisations, their applicability to data glyphs has not been fully explored. Data glyphs are composite visual entities that encode multiple data dimensions within a single graphical object and may employ heterogeneous visual elements, mapping strategies, and internal layouts. This work investigates whether the GoG can serve as a foundation for the grammar-based generation of data glyphs. We conduct a theoretical analysis of the GoG layers and apply them to two representative glyphs with distinct structural characteristics: a star glyph and a Chernoff face. The comparison indicates that the star glyph can be described through a relatively direct application of the original layered pipeline. In contrast, the composite structure of the Chernoff face requires component-specific mappings and suggests the need for an iterative application of the graph layers followed by an explicit graphical composition mechanism. These findings indicate that the GoG provides a useful foundation for describing data glyph construction but may require adaptations to support heterogeneous components and complex internal layouts. This work constitutes an initial theoretical investigation and provides directions for future grammar-based approaches to data glyph generation.

Index Terms: Grammar of Graphics, Data Glyphs, Grammar-Based Visualization, Multivariate Visualization, Glyph Generation.

1 INTRODUCTION

The Grammar of Graphics (GoG) [16] provides a formal and compositional framework for describing how data are transformed into graphical representations. Rather than defining visualisations as a collection of predefined chart types, the GoG decomposes visualisation construction into independent layers, including data, variables, algebra, scales, statistics, geometry, coordinates, and aesthetics. This perspective has significantly influenced visualisation research and the development of grammar-based visualisation systems, supporting the systematic specification, generation, and composition of visual representations.

Data glyphs are composite graphical objects that encode multiple data dimensions within a single visual entity [1]. By combining several visual variables, glyphs can support the representation of multivariate data while preserving the identity of individual observations. However, the complexity and heterogeneity of glyph structures may present challenges for existing visualisation grammars. In particular, different data dimensions may be mapped to distinct graphical components, each with its own scale, geometry,

coordinate configuration, and visual encoding. These characteristics raise the question: *Can the Grammar of Graphics (GoG) serve as a foundational framework for generating data glyphs?*

To address this question, we conduct a theoretical analysis in three phases. First, we examine the layers and operations defined by the GoG to establish their roles in data glyph design. Second, we apply the grammar to two data glyphs with different structural and mapping characteristics: a star glyph and a Chernoff face. Finally, we compare the resulting specifications to identify limitations and potential adaptations required to represent composite glyph structures. The analysis focuses on the conceptual specification of the glyphs; the implementation and empirical evaluation of the resulting visualisations are outside the scope of this work.

2 BACKGROUND

2.1 Data Glyphs

Data glyphs are composite graphical objects that encode multiple data dimensions within a single visual entity [1, 3]. By combining multiple visual variables, glyphs represent multivariate observations while preserving data identity. Examples include arrows encoding directional information [17], star glyphs mapping magnitudes along radial axes [14], and Chernoff faces encoding attributes through facial features [2].

Data glyph designs vary significantly, depending on their application domain, data mapping, and layout. Their mappings may be *one-to-one*, in which individual data dimensions are associated with distinct visual elements, or *many-to-one*, where dimensions share the same mapping logic. Glyphs can be *pictorial* or *non-pictorial*, using abstract or metaphoric relationships.

Internally, elements can be juxtaposed (side-by-side) or superimposed (overlapped), in 2D or 3D space across various layout patterns (e.g., grids or radial arrangements) using visual variables.

This variety of designs offers flexibility but makes it difficult to describe and generate data glyphs systematically. In particular, different glyph structures may require distinct combinations of data transformations, visual mappings, geometric components, and spatial layouts. These characteristics motivate the investigation of grammar-based approaches as a potential foundation for formally describing and generating data glyphs.

2.2 Grammar-based approaches

The Grammar of Graphics (GoG) provides a foundational, compositional framework for specifying statistical graphics by decomposing them into independent components—such as data, transformations, scales, geometries, and coordinate systems [16]. This declarative approach contrasts with template-based systems, offering greater flexibility and reusability. GoG principles have been widely adopted in modern visualisation systems, most notably in *ggplot2* for R [15] and the Vega/Vega-Lite ecosystems for the web [13], which facilitate extensible visualisation authoring.

Research has extended the GoG to address requirements beyond standard charts. These adaptations support temporal and animated data (e.g., *ggtime* [5], Animated Vega-Lite [18]), uncertainty (PGoG [10], *ggdist* [7]), and specialized layouts like mosaic

*e-mail: barbararamos@dei.uc.pt

†e-mail: jmacunha@dei.uc.pt

‡e-mail: evgheni@dei.uc.pt

plots [6] and unit visualizations [9]. Further, studies on the cognitive and practical aspects of these grammars highlight that their effectiveness relies on how well their abstractions align with users' mental models and authoring workflows [11, 12].

Despite the popularity of grammar-based systems, we still do not fully understand how to apply them to data glyphs. While algebraic processes for visualisation offer relevant conceptual insights [8], no existing work systematically investigates how GoG layers and operators can describe the construction of multivariate, composite glyphs. These entities often require complex, multi-layered coordinate systems and heterogeneous mapping strategies, motivating the analysis presented in this work.

3 THE GRAMMAR OF GRAPHICS FOR DATA GLYPHS

The Grammar of Graphics (GoG) decomposes visualisation into a sequence of layers that transform data into graphical elements [16]. This layered structure serves as a foundation for generating data glyphs. However, because glyphs are composite objects encoding multiple dimensions within a single entity, some GoG components apply directly, while others may require adaptation. We examine each layer below.

3.1 Data

The Data layer defines the source from which observations and attributes are obtained. Data may be stored in databases, files, or other structured sources and constitute the input to the graphical pipeline. At this stage, the data are not yet associated with graphical properties or transformed into visual representations.

Glyph-based visualisations can use the same data sources and observational structures as conventional visualisations. Each observation may represent an individual entity, while its associated attributes provide the values that will subsequently be encoded by the glyph. Thus, the Data layer applies directly to data glyphs.

3.2 Variables

This layer selects the attributes to be visualised. For data glyphs, it also establishes the dimensions represented and whether they follow a *one-to-one* or *many-to-one* mapping strategy, which directly governs subsequent scales, geometries, and aesthetics. Complex glyphs with independent component mappings may therefore require explicitly linking variables to specific glyph sub-components.

Thus, the Variables layer remains compatible with data glyphs but may require an explicit description of the relationship between data dimensions and glyph components. This relationship becomes particularly relevant for complex glyphs in which different visual elements use independent mappings.

3.3 Algebra

Algebra combines variables into a *varset*, which depends on the structure of the input data rather than the visual complexity of the glyph. Algebraic operations are necessary only if the data require combining or restructuring before mapping.

3.4 Scales

The Scales layer transforms variable values into domains usable by subsequent graphical operations. Scales play a central role in data glyphs because the dimensions represented by a glyph may have different units, ranges, and distributions. In *many-to-one* mappings, where multiple dimensions share the same graphical logic, the variables commonly require normalisation to a common output range. This allows their values to be represented using comparable graphical parameters. In contrast, *one-to-one* mappings may require independent scale functions because each dimension is associated with a different visual element or geometric parameter.

3.5 Statistics

The Statistics layer applies statistical methods to the varset to produce graphical summaries or transformations. For data glyphs, statistical transformations depend on the intended unit of representation. When each glyph represents an individual observation, no statistical aggregation may be required, and the original observations can be directly mapped to glyphs. Alternatively, statistical methods may be used when a glyph represents an aggregate, such as the mean or distribution of a group of observations.

Therefore, the Statistics layer does not require a glyph-specific modification. Its use is determined by whether glyphs represent individual observations or statistical summaries.

3.6 Geometry

The Geometry layer defines the graphical structures used to represent the transformed data. Geometric operations generate graphical marks from the varset, such as points, lines, intervals, areas, or other graphical objects. At this stage, the data are converted into geometric representations, although subsequent layers define their final spatial arrangement and visual appearance.

For data glyphs, geometry defines the visual elements that encode the data dimensions. In glyphs based on *many-to-one* mappings, a common geometric structure may be repeated for multiple dimensions. In *one-to-one* mappings, different dimensions may control distinct geometric primitives, resulting in a composite glyph formed by multiple heterogeneous elements.

This layer introduces one of the main challenges in applying the GoG to data glyphs. Conventional geometries are generally designed to generate a limited set of graphical marks from observations. Complex glyphs may instead require the construction and composition of multiple geometries within a single graphical element. Consequently, glyph construction may require an extended geometric model capable of representing composite and parametrised graphical structures.

3.7 Coordinates

The Coordinates layer transforms the spatial representation produced by the Geometry layer. Coordinate transformations define how graphical elements are positioned within a frame and may convert geometric structures between different coordinate systems.

For data glyphs, the Coordinates layer can be interpreted as defining the internal layout of glyph components. In this context, coordinate systems determine how the graphical elements generated by the Geometry layer are arranged relative to one another. Radial glyphs, for example, may use polar coordinates to distribute dimensions around a central point, whereas other glyphs may use Cartesian coordinates or predefined local reference positions.

Thus, the Coordinates layer provides a conceptual basis for glyph layouts. However, complex glyphs may require local coordinate systems, nested coordinate spaces, or component-specific transformations. These requirements may exceed the use of a single global coordinate transformation and suggest the need for a more explicit representation of internal glyph layouts.

3.8 Aesthetics

The Aesthetics layer translates graphical structures into visual representations by mapping data to visual properties like colour or size. Data glyphs rely extensively on aesthetic mappings because multiple dimensions are encoded through visual properties. However, complex glyphs may contain several graphical elements, each with an independent mapping logic. In *one-to-one* glyphs, different dimensions may control different aesthetic properties or may use distinct aesthetic functions for separate components. In *many-to-one* glyphs, multiple dimensions may share the same aesthetic logic.

4 EXAMPLES

This section applies the Grammar of Graphics (GoG) to two multivariate glyphs using the Palmer Penguins dataset [4]: a star glyph and a Chernoff face. The examples were selected because they represent distinct approaches to glyph construction. The star glyph uses a homogeneous radial structure, whereas the Chernoff face combines multiple heterogeneous graphical components.

4.1 Common Varset Specification

Both examples use the same data, where each observation represents a penguin described by four variables: bill length (B_L), bill depth (B_D), flipper length (F_L), and body mass (M). The selected varset is defined as $V = \{B_L, B_D, F_L, M\}$. As these variables are already organised as multivariate observations, no cross, blend, or nest operation is required. The Algebra layer preserves the varset, $A(V) = V$. Similarly, since each glyph represents a single observation, no statistical transformation is needed, so the Statistics layer also preserves the data.

The examples differ mainly in their Scales, Geometry, Coordinates, and Aesthetics. The star glyph applies a shared mapping logic to all variables, whereas the Chernoff face uses component-specific mappings.

4.2 Star Glyph

A star glyph represents each quantitative variable along a radial axis. As the variables have different units and ranges, they are normalised to a common interval using min–max scaling:

$$s_i(v_i) = \frac{v_i - \min(V_i)}{\max(V_i) - \min(V_i)}, \quad s_i(v_i) \in [0, 1]. \quad (1)$$

The variables are ordered as $\pi = (B_L, B_D, F_L, M)$, which defines their positions around the glyph. Each scaled value determines the radial distance of a vertex. The vertices are placed at equally spaced angular positions and connected to form a closed polygon:

$$G_{star} = \text{polygon}(p_1, p_2, p_3, p_4), \quad p_i = (r_i \cos \theta_i, r_i \sin \theta_i), \quad (2)$$

where $r_i = s_i(v_i)$ and $\theta_i = 2\pi(i-1)/4$. The species variable is mapped to the polygon colour, while fill, stroke, and opacity remain fixed visual properties.

4.3 Chernoff Face

Chernoff faces [2] represent multivariate observations through facial features. Unlike the star glyph, each variable is assigned to a distinct facial parameter and therefore requires a component-specific scale: $S_{face} = \{s_{B_L}, s_{B_D}, s_{F_L}, s_M\}$.

The scaled values control the parameters of the head, eyes, nose, and mouth. The glyph is therefore composed of multiple geometric primitives, $G_{face} = G_{head} \cup G_{eyes} \cup G_{nose} \cup G_{mouth}$, arranged within a local Cartesian coordinate system using predefined positions relative to the centre of the face. The corresponding scale functions determine their dimensions and shapes. As in the star glyph, species is mapped to colour, while fill, stroke, and opacity define the appearance of the components without encoding additional data.

5 DISCUSSION

The comparison summarized in Figure 1 demonstrates that the Grammar of Graphics (GoG) can describe the fundamental data transformations and visual mappings involved in data glyph construction. The varset layers were compatible with both examples, requiring no substantial adaptation. The main limitations emerged in the graph layers, particularly in the relationship between scales, geometry, and coordinates.

The star glyph follows a homogeneous structure in which all variables share the same mapping logic. The variables are normalised to a common range, positioned using a shared polar coordinate system, and represented as vertices of a single polygon. Consequently, the glyph can be generated through a single application of the GoG pipeline. In contrast, the Chernoff face is a heterogeneous composite object. Each variable controls a distinct facial component and may require a specific scale, geometry, and local coordinate configuration. The final glyph is then obtained by combining these components.

This difference reveals a limitation in interpreting the GoG as a strictly linear pipeline. While a single sequence of graph layers is sufficient for the star glyph, the Chernoff face requires the *scale* $\rightarrow$ *geometry* $\rightarrow$ *coordinates* $\rightarrow$ *aesthetics* sequence to be applied separately to multiple components. The construction can therefore be represented as a component-wise process:

$$c_k = A_k(C_k(G_k(S_k(v_k)))), \quad \mathcal{G} = \bigoplus_{k=1}^K c_k, \quad (3)$$

where S_k , G_k , C_k , and A_k define the scale, geometry, coordinates, and aesthetics of component c_k , respectively, and $\bigoplus$ denotes the composition of the components into the final glyph. This interpretation is better described as *iterative*, *recursive*, or *component-wise* rather than a linear sequence of the pipeline. Instead, the graph layers are reused to construct different components.

The Coordinates layer presents a related limitation. The star glyph uses a single polar coordinate system, whereas the Chernoff face relies on the relative placement of multiple components within a local spatial structure. Complex glyphs may therefore require multiple local or nested coordinate spaces. This suggests that the notion of *glyph layout* could extend the Coordinates layer by explicitly defining the spatial organisation and relationships among glyph components.

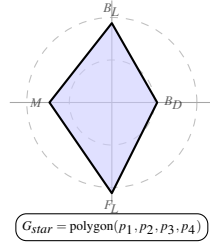
The Aesthetics layer was largely consistent across both examples. However, complex glyphs may require component-specific visual properties, such as different fills or strokes for individual components. When these properties do not encode data, they should be treated as fixed styling parameters rather than data-driven aesthetics. An adaptation for data glyphs may therefore need to distinguish data-driven visual mappings from component-level styling.

Overall, the examples suggest that the GoG provides a useful foundation for describing data glyphs but may require adaptations to represent complex composite structures. Such adaptations could preserve the existing varset layers while extending the graph layers with component-specific mappings, local coordinate systems, and an explicit composition or layout operation.

6 CONCLUSION

This work investigated the applicability of the Grammar of Graphics (GoG) to data glyph construction through the theoretical analysis of a star glyph and a Chernoff face. The examples demonstrated that the GoG provides a useful framework for describing the data transformations and visual mappings involved in glyph construction. However, while the star glyph can be represented using a relatively direct application of the original layered pipeline, the composite structure of the Chernoff face suggests the need for a component-wise application of the graph layers and an explicit mechanism for combining graphical components.

This analysis represents an initial theoretical investigation rather than a formal extension or validation of the GoG. Future work should implement these representations using existing grammar-based visualization libraries, evaluate a broader range of glyph types, and investigate the proposal of a grammar-based adaptation specifically designed to support data glyph design.



DATA: $\mathbf{D} = (B_L, B_D, F_L, M, \text{Species})$

VARIABLES: $\mathbf{V} = \{B_L, B_D, F_L, M\}$

ALGEBRA: $A(\mathbf{V}) = \mathbf{V}$

SCALE:

$$s_i(v_i) = \frac{v_i - \min(V_i)}{\max(V_i) - \min(V_i)}, \quad s_i(v_i) \in [0, 1]$$

STATISTICS: $\text{statid}(\mathbf{V}) = \mathbf{V}$

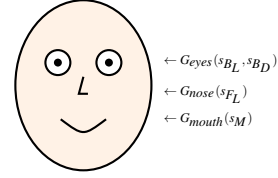
GEOMETRY:

$$G_{\text{star}} = \text{polygon}(p_1, p_2, p_3, p_4)$$

AESTHETICS:

$$A_{\text{star}} = \{\text{colour}(\text{Species}), \text{fill}, \text{stroke}, \text{opacity}\}$$

Star Glyph



DATA: $\mathbf{D} = (B_L, B_D, F_L, M, \text{Species})$

VARIABLES: $\mathbf{V} = \{B_L, B_D, F_L, M\}$

ALGEBRA: $A(\mathbf{V}) = \mathbf{V}$

SCALE:

$$S_{\text{face}} = \{s_{B_L}, s_{B_D}, s_{F_L}, s_M\}$$

STATISTICS: $\text{statid}(\mathbf{V}) = \mathbf{V}$

GEOMETRY:

$$G_{\text{face}} = G_{\text{head}} \cup G_{\text{eyes}} \cup G_{\text{nose}} \cup G_{\text{mouth}}$$

AESTHETICS:

$$A_{\text{face}} = \{\text{colour}(\text{Species}), \text{fill}, \text{stroke}, \text{opacity}\}$$

Chernoff Face

Figure 1: Grammar of Graphics specifications for the star glyph and the Chernoff face using the Palmer Penguins dataset.

ACKNOWLEDGMENT

This work is funded by national funds through FCT – Foundation for Science and Technology, I.P., within the scope of the research unit UID/00326 - Centre for Informatics and Systems of the University of Coimbra, and supported by the project Autoglyphs: Automatic Approaches for Generative Design of Data Glyphs. The first author was supported by FCT – Foundation for Science and Technology, I.P., through the PhD grant 2023.02333.BD.

REFERENCES

- [1] R. Borgo, J. Kehrner, D. H. S. Chung, E. Maguire, R. S. Laramée, H. Hauser, M. Ward, and M. Chen. Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. In M. Sbert and L. Szirmay-Kalos, eds., *Eurographics 2013 - State of the Art Reports*. The Eurographics Association, 2013. doi: /10.2312/conf/EG2013/stars/039-063 1
- [2] H. Chernoff. The Use of Faces to Represent Points in k-Dimensional Space Graphically. *Journal of the American Statistical Association*, 68(342):361–368, June 1973. doi: 10.1080/01621459.1973.10482434 1, 3
- [3] J. Fuchs, P. Isenberg, A. Bezerianos, and D. Keim. A systematic review of experimental studies on data glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 23(7):1863–1879, July 2017. doi: 10.1109/tvcg.2016.2549018 1
- [4] A. M. Horst, A. P. Hill, and K. B. Gorman. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*, 2020. R package version 0.1.0. doi: 10.5281/zenodo.3960218 3
- [5] C. A. Huang, M. O’Hara-Wild, R. J. Hyndman, and M. Kay. Ggtime: A Grammar of Temporal Graphics, 2025. doi: 10.48550/ARXIV.2510.25656 1
- [6] H. Jeppson and H. Hofmann. Generalized Mosaic Plots in the ggplot2 Framework. *The R Journal*, 14(4):50–78, Feb. 2023. doi: 10.32614/RJ-2023-013 2
- [7] M. Kay. Ggdist: Visualizations of Distributions and Uncertainty in the Grammar of Graphics. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2023. doi: 10.1109/TVCG.2023.3327195 1
- [8] G. Kindlmann and C. Scheidegger. An Algebraic Process for Visualization Design. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2181–2190, Dec. 2014. doi: 10.1109/TVCG.2014.2346325 2
- [9] D. Park, S. M. Drucker, R. Fernandez, and N. Elmqvist. ATOM: A Grammar for Unit Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3032–3043, Dec. 2018. doi: 10.1109/TVCG.2017.2785807 2
- [10] X. Pu and M. Kay. A Probabilistic Grammar of Graphics. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13. ACM, Honolulu HI USA, Apr. 2020. doi: 10.1145/3313831.3376466 1
- [11] X. Pu and M. Kay. How Data Analysts Use a Visualization Grammar in Practice. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–22. ACM, Hamburg Germany, Apr. 2023. doi: 10.1145/3544548.3580837 2
- [12] X. Pu, M. Kay, S. M. Drucker, J. Heer, D. Moritz, and A. Satyanarayan. Special Interest Group on Visualization Grammars. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–3. ACM, Yokohama Japan, May 2021. doi: 10.1145/3411763.3450406 2
- [13] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, Jan. 2017. doi: 10.1109/TVCG.2016.2599030 1
- [14] J. H. Siegel, E. J. Farrell, R. M. Goldwyn, and H. P. Friedman. The surgical implications of physiologic patterns in myocardial infarction shock. *Surgery*, 72(1):126–141, 1972. 1
- [15] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. 1
- [16] L. Wilkinson. *The Grammar of Graphics*. Statistics and Computing. Springer, Berlin, 2 ed., 2005. doi: 10.1007/0-387-28695-0 1, 2
- [17] C. Wittenbrink, A. Pang, and S. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):266–279, 1996. doi: 10.1109/2945.537309 1
- [18] J. Zong, J. Pollock, D. Wootton, and A. Satyanarayan. Animated Vega-Lite: Unifying Animation with a Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):149–159, Jan. 2023. doi: 10.1109/TVCG.2022.3209369 1