1 Discussion

This work contributes a mathematical description of the transformation from data to visual representation. Combining Butler's fiber bundle model of data with Spivaks formalism of data schemas provides a way of decoupling topology from variability such that the model can support a very large variety of datasets, including discrete relational tables, multivariate high resolution spatio temporal datasets, and complex networks. Modeling the graphic as a fiber bundle provides a way to separate the target display space from the topology of the graphic. By decomposing the mapping from data to visual representation as encoding ν, assembly Q, and a mapping between data and graphic topologies $\xi$and formalizing what equivariance each stage needs to preserved, this work derives constraints that visualization library authors could embed in their code to guarantee visualizations that are equivariant transforms of the input data.

This work generalizes previous research constraining visual encodings from data components to graphic components as equivariant maps to components that are N-dimensional. Furthermore, it precisely defines the glyph as the visual element constructed from data on a simplex in a simplacial complex where the simplex is discrete or continuous. This is a restatement of for example Bertin's definition of a line that encapsulates all points on the continuous line. By modeling the data topology along with its variablity, this work also provides a generalization of topology preservation as a deformation retraction from graphic space to data space. By using a functional paradigm, we can deconstruct the graphic to the glyph associated with each data point or even to pieces of a glyph; therefore the renderer has full flexibility in how to to generate the image, while the graphic to data topology maps provide a way to keep track of which part of the image belongs to which data point.

The toy prototype built using this model validates that is usable for a general purpose visualization tool since it can be iteratively integrated into the existing architecture rather than starting from scratch/ Factoring out glyph formation into assembly functions allows for much more clarity in how the glyphs differ. This prototype demonstrates that this framework can generate the fundemental marks, point (scatter plot), line (line chart), and area (bar chart). Furthermore, the grouped and stacked bar examples demonstrate that this model supports composition of glyphs into more complex graphics. These composite examples also rely on the fiber bundles section base book keeping to keep track of which components contribute to the attributes of the glyph. Implementing this example using a Pandas dataframe demonstrates the ease of incorporating existing widely used data containers rather than requiring users to conform to one stands.

1.1 Limitations

So far this model has only been worked out for a single data set tied to a primitive mark, but it should be extensible to compositing datasets and complex glyphs. The examples and prototype have so far only been implemented for the static 2D case, but nothing in the math limits to 2D and expansion to the animated case should be possible because the model is formalized in terms of the sheaf. While this model supports equivariance of figurative glyphs generated form parameters of the data[1, 4], it currently does not have a way to evaluate the semantic accuracy of the figurative representation. This model also does not currently factor in effectiveness, but potentially effectiveness criteria could be incorporated into a scheme for assigning encoders and assembling glyphs. Also, even though the model is designed to be backend independent, it has only really been tested against the AGG backend. It is especially unknown how this framework interfaces with high performance rendering libraries such as openGL[5]. Because this model has been limited to the graphic design space, it does not address the critical task of laying out the graphics in the image

This model and the associated prototype is deeply tied to Matplotlib's existing architecture. While the model is expected to generalize to other libraries, such as those built on Mackinlay's APT framework, this has not been worked through. In particular, Mackinlay's formulation of graphics as a language with semantic and syntax lends itself a declarative interface[12], which Heer and Bostock use to develop a domain specific visualization language that they argue makes it simpler for designers to construct graphics without sacrificing expressivity [9]. Similarly, the model presented in this work formulates visualization as equivariant maps from data space to visual space, and is designed such that developers can build software libraries with data and graphic topologies tuned to specific domains.

1.2 Future Work

While the model and prototype demonstrate that generation of simple marks from the data, there is a lot of work left to develop a model that underpins a minimally viable library. Foremost is implementing a data object that encodes data with a 2D continous topology and an artist that can consume data with a 2D topology to visualize the image[7, 8, 16] and also encoding a separate heatmap[11, 18] artist that consumes 1D discrete data. A second important proof of concept artist is a boxplot[17] because it is a graphic that assumes computation on the data side and the glyph is built from semantically defined components and a list of outliers.

The model supports simple composition of glyphs by overlaying glyphs at the same position, but more work is needed to define an operator where the fiber bundles have shared $S_2 \hookrightarrow S_1$ such that fibers could be pulled back over the subset. This complex operator is necessary for building semantically meaningful components such as interactive visualizations that update on shared S, such as brush-linked views[2, 3] and verifying constraints in multiple view systems [14]. The models simple addition supports axes as standalone artists with overlapping visual position encoding, but the complex operator would allow for keeping labels consistent with text when for example the horizontal and vertical positions of the data are changed. In summary, the proposed scope of work for the dissertation is

- expansion of the mathematical framework to include complex addition
- mathematical formulation of a graphic with axes labeling
- implementation of data with 2D continuous topology and a compatible artist
- provisional mathematics and implementation of user level composite artists
- proof of concept domain specific user facing library

Additionally, implementing the complex addition operator would allow for developing a mathematical formalism and prototype of how interactivity would work in this model. Other potential tasks for future work is implementing a data object for a non-trivial fiber bundle and exploiting the models section level formalism to build distributed data source models and concurrent artists. This could be pushed further to integrate with topological[10] and functional [15] data analysis methods. While this paper formulates visualization in terms of monoidal action homomorphisms between fiberbundles, the model lends itself to a categorical formulation[6, 13] that could be further explored.

2 Conclusion

Despite the alternative formalism to Mackinlay's APT framework, it is indebted to APT for providing a guideline for issues the model needs to address. As with APT, this works aims to develop a framework on which to build visualization tools. Unlike APT, our model is geared towards balancing the needs of the the scientific and information visualization communities; therefore we present a framework that can describe both how the components of the data are encoded and how the continuity is preserved in the graphic. We hope that this framework can distill the constraints on the data to graphic mapping such that it can be used to improve the architecture of a heavily used visualization tool and allow developers to more easily build domain specific tools.