

1 Discussion

This work contributes a functional model of the structure-preserving maps from data to visual representation to guide the development of visualization libraries, thereby providing a means to express the constraints that visualization must preserve continuity and faithfully translate the properties of the data variables into visual variables. Combining Butler’s proposal of a fiber bundle model of visualization data with Spivak’s formalism of schema lets this model support a variety of datasets, including discrete relational tables, multivariate high resolution spatio temporal datasets, and complex networks. By decomposing the artist into encoding ν , assembly Q , and reindexing ξ maps, the model expresses the specifications that graphic and data must have equivalent *continuity* equivalent to the data, and that the visual characteristics of the graphics are *equivariant* to their corresponding components under monoid actions. TEAM defines these constraints on the transformation function such that they are not specific to any one type of encoding or visual characteristic. The toy prototype built using this model validates that it is usable for a general purpose visualization tool since it can be iteratively integrated into existing architecture rather than starting from scratch. Factoring out graphic formation into assembly functions allows for much more clarity in how they differ. This prototype demonstrates that this framework can generate the fundamental point (scatter plot) and line (line chart) marks.

1.1 Limitations

Our model and prototype are deeply tied to Matplotlib’s existing architecture, so it has not yet been worked through how the model generalizes to libraries such as VTK or D3. Even though the model is designed to be backend and format independent, it has only been tested using PNGs rendered with AGG[1]. It is unknown how this framework interfaces with high performance rendering libraries such as OpenGL[2] that implement different models of ρ . While our model supports equivariance of figurative glyphs [3] generated from data components[4, 5], it cannot evaluate the semantic accuracy of the figurative representation. Effectiveness criteria[6, 7] are out of scope.

1.2 Future Work

More work is needed to formalize the composition operators, equivalence class A' , and the mathematical model of interactivity. We also need to implement artists that demonstrate that the model can underpin a minimally viable library, foremost an image[8, 9], a heatmap[10, 11], and an inherently computational artist such as a boxplot[12]. In summary, the optimistic proposed scope of work is

work period	milestones & tasks
April-July 2021	prepare conference presentation on new functionality enabled by model for <i>SciPy</i> : artists that do not inherit from existing Matplotlib artists, computational artists such as histograms, non tabular data, composite interactive artist
June-September 2021	prepare theory paper on interactivity for submission to <i>TCVG or Eurovis 2022</i> : fully work out and describe math of addition operators and lookups from graphic to data space, implement brush linked artist (shared base space) and artist that exploits sheafs
May - November 2021	prepare applications paper on high dimensional data for submission to <i>TCVG</i> : math and implementation of computational artists, concurrent artists and data sources, non-trivial data bundles
August 2020-February 2022	prepare systems paper on building domain specific libraries based on this model for <i>Infviz 2022</i> : domain specific structured data, composite artists, inference of meta data components, mathematical notion of a visualization (labeled, multiple artists, etc)
March 2022	dissertation writing: synthesize previous work on climate data, compile topological equivariant artist model work
April 2022	defense

Table 1

34 In acknowledgement that the schedule is optimistic, the data applications of this work
 35 could instead be revisited after the defense as that work is not integral to visualization
 36 library architecture. The data applications could be further integrated with topological[13]
 37 and functional [14] data analysis methods. Since this model formalizes notions of structure
 38 preservation, it can serve as a good base for tools that assess quality metrics[15] or invariance
 39 [16] of visualizations with respect to graphical encoding choices. While this paper formulates
 40 visualization in terms of monoidal action homomorphisms between fiberbundles, the model
 41 lends itself to a categorical formulation[17, 18] that could be further explored.

42 2 Conclusion

43 A TEAM driven refactor of visualizations libraries could produce more maintainable,
 44 reusable, and extensible code, leading to better building blocks for the ecosystem of tools

45 built on top of TEAM architected libraries. Building block libraries could better support
46 downstream, including domain specific, libraries without having to explicitly incorporate
47 the specific data structure and visualization needs of those domains back into the base
48 library. Adopting this model would induce a separation of data representation and visual
49 representation that, for example, in Matplotlib is so entangled that it has lead to a brittle
50 and sometimes incoherent API and internal code base. A refactor that incorporated
51 the generalized data model and functional transforms presented in TEAM would lead to
52 building block libraries that provide a more consistent, reusable, flexible, collection of
53 blocks.