Topological Equivariant Artist Model for Visualization Library Architecture

Hannah Aizenman, Thomas Caswell, and Michael Grossberg, *Member, IEEE,*

Abstract—The abstract goes here.

Index Terms—

1 Introduction

This paper uses methods from topology and category theory to develop a model of the transformation from data to graphical representation. This model provides a language to specify how data is structured and how this structure is carried through in the visualization, and serves as the basis for a functional approach to implementing visualization library components. Topology allows us to describe the structure of the data and graphics in a generalizable, scalable, and trackable way. Category theory provides a framework for separating the transformations implemented by visualization libraries from the various stages of visualization and therefore can be used to describe the constraints imposed on the library components [**?**], [**?**]. Well constrained modular components are inherently functional [**?**], and a functional framework yields a library implementation that is likely to be shorter, clearer, and more suitd to distributed, concurrent, and on demand tasks [**?**]. Using this functional approach, this paper contributes a practical framework for decoupling data processing from visualization generation in a way that allows for modular visualization components that are applicable to a variety of data sets in different formats. is it OK that this is something reviewer 4 wrote

2 Related Work

This work aims to develop a model for describing visualization transformations that can serve as guidance for how to architecture a general purpose visualization library. We define a general purpose visualization library as one that provides non domain specific building block components [**?**] for building visualizations, for example functions for converting data to color or encoding data as dots. In this section, we describe how visualization libraries attempt this goal and discuss work that formally describes what properties of data should be preserved in a visualization. We restrict the properties of data that should be preserved to

- *H. Aizenman and M. Grossberg are with the department of Computer Science, City College of New York.*
  E-mail: haizenman@ccny.cuny.edu, mgrossberg@ccny.cuny.edu
- *Thomas Caswell is with National Synchrotron Light Source II, Brookhaven National Lab*
  E-mail: tcaswell@bnl.gov

Manuscript received X XX, XXXX; revised X XX, XXXX.

continuity how elements in a dataset are connect to each other, e.g. discrete rows in a table, networked nodes, pixels in an image, points on a line

equivariance functions on data that have an equivalent effect on the graphical representation, e.g. rotating a matrix has a matching rotation of the image, translating the points on a line has a matching visual shift in the line plot

2.1 Continuity

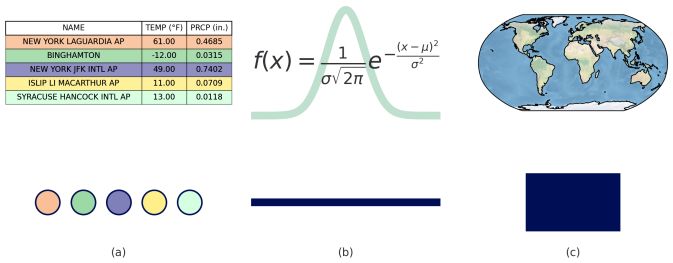


Fig. 1: Continuity is how elements in a data set are connected to each other, which is distinct from how the data is structured. The rows in (a) are discrete, therefore they have discrete continuity as illustrated by the discrete dots. The gaussian in (b) is a 1D continuous function, therefore the continuity of the elements of the gaussian can be represented as a line on an interval (0,1). In (c), every element of the globe is connected to its nearest neighbors, which yields a 2D continuous continuity as illustrated by the square.

Continuity is a representation of how the elements in a dataset are connected to each other. For example, in **??**, each station record in the table is independent of the others; therefore, the continuity of the table is discrete. The data provided by the gaussian are points sampled along the curve, therefore the continuity of the points on the line is 1D continuous. Every point on the globe is connected to its 6 nearest cardinal neighboring points (NW, N, NE, E, SE, S, SW, W).

Often continuity is expressed in the choice of visual algorithm (visualization type), as explored in taxonomies by Tory and Möller [**?**] and Chi [**?**]. For example, in **??** the same table can be interpreted as a set of 1D continuous curves

Fig. 2: Continuity is implicit in choice of visualization rather than explicitly in choice of data container. The line plots in (b) are generated by a 2D table (a). Structurally this table can be identical to the 2D matrix (a) that generates the image in (c).

when visualized as a collection of line plots or as a 2D surface when visualized as an image. This means that often there is no way to express data continuity independent of visualization type, meaning most visualization libraries will allow, for example, visualizing discrete data as a line plot or an image. General purpose visualization libraries-such as Matplotlib [?], Vtk [?], [?], and D3 [?]-carry distinct data models as part of the implementation of each visual algorithm. The lack of unified data model means that each plot in a linked [?], [?] visualization is treated as independent, as are the transforms converting each field in the data to a visual equivalent.

Domain specific libraries can guarantee consistency because they have a single model of the data in their software design, as discussed in Heer and Agarwal [?]'s survey of visualization software design patterns. For example, the relational database is core to tools influenced by APT, such as Tableau [?], [?], [?] and the Grammar of Graphics [?] inspired ggplot [?], Vega [?] and Altair [?]. Images underpin scientific visualization tools such as Napari [?] and ImageJ [?] and the digital humanities oriented ImagePlot [?] macro; the need to visualize and manipulate graphs has spawned tools like Gephi [?], Graphviz [?], and Networkx [?].

2.1.1 *Fiber Bundles*

A model that allows for expressing data continuity in a general flexible way allows us to express, and therefore be faithful to, the multivariate consistency constraints described by Qu and Hullman [?]. A general data model also allows for consistent adaptation to modern data needs, such as complex, metadata rich, distributed, or streaming data. The mathematical theory of fiber bundles provides one such abstraction that can express complicated dimensionality and continuity without being tied to any one data container type, as proposed by Butler, Bryson, and Pendley [?], [?]. In this paper, we build on their work that proposes using topological spaces to represent different properties of data.

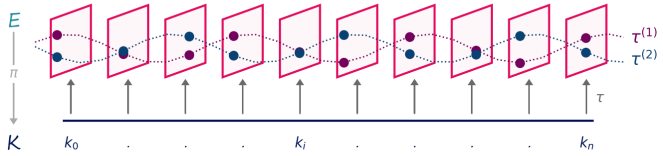


Fig. 3: A fiber bundle is mathematical construct that allows us to express the relationship between data and continuity. The total space Eis the topological space in which the data is embedded. The fiber space F is embedded in E and is the set of all possible values that any add big rectangle E

Here we present a brief summary of topology, for more information see Hatcher [?], Munkres [?], and Bradley et. al. [?].

A topological space a topological space $(X, \mathcal{T})$ is a set X with a topology $\mathcal{T}$. Topologies are collections of open sets

such that the empty set and X are in the collection of open sets $\mathcal{T}$, the union of elements in $\mathcal{T}$

Specifically, Butler, Bryson, and Pendley suggest that fiber bundles be the basis of an abstract data model. Fiber bundles are a collection (E, K, F, π) of topological spaces

Total Space E
Fiber Space F
Base Space K

with a projection map $\pi : E \to K$ that connects every point in E to a point in K.

$$F \hookrightarrow E \xrightarrow{\pi} K \tag{1}$$

As indicated by $\hookrightarrow$, the fiber space F is embedded inside the total space EThis is illustrated in **??**, wherein values lives in the fiber $F \subseteq E$. In this example, the fiber F is the cartesian product of two sets $F_0 \times F_1$ where each fiber F_i is ....

$$\mathrm{formalequationofthefiber?} \tag{2}$$

While the values are embedded in F, the continuity of the data is modeled as the base space K, which is the quotient space [?]

$$\mathrm{formalmathofthebasespaceasequivalenceclassforF_{[}k]} \tag{3}$$

which means that every point in F_k maps to a point $k \subset K$.

$$\begin{array}{ccc} F & \hookrightarrow & E \\ & & \pi \downarrow \uparrow \tau \\ & & K \end{array} \tag{4}$$

2.2 Sheaf Maps $\mathcal{O}$

We can use sheafs to ensure continuity even when the data is broken up. Is the glue rules -¿ we can haz parallism.

$$A : \mathcal{O}(E) \to \mathcal{O}(H) \tag{5}$$

2.3 Equivariance

What is it?

$$\begin{array}{ccccc} \mathrm{data} & \longrightarrow & \mathrm{representation} & \longrightarrow & \mathrm{visual\ stimulus} \\ \downarrow \mathrm{function} & & & & \downarrow \mathrm{visual\ equivalent\ to\ fu} \\ \mathrm{data} & \longrightarrow & \mathrm{representation} & \longrightarrow & \mathrm{visual\ stimulus} \end{array}$$

Fig. 4

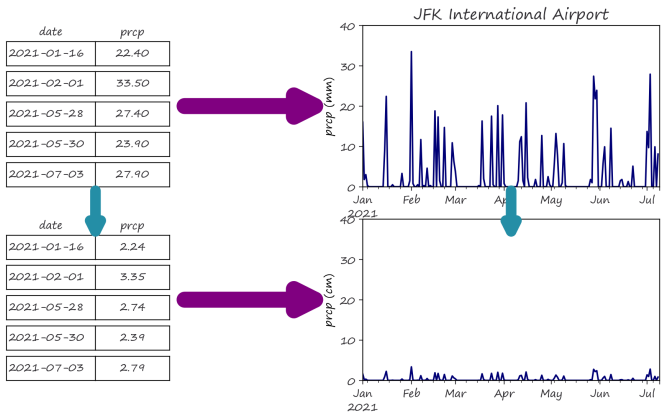


Fig. 5: Equivariance is that a transformation on the data has a corresponding transformation in the graphical representation. For example, in this figure the data is scaled by a factor 10. Equivalently the line plot is scaled by factor of 10, resulting in a shrunken line plot. Either a transformation on the data side can induce a transformation on the visual side, or a transformation on the visual side indicates that there is also a transformation on the data side.

2.3.1 *Category Theory*

In this work, we propose that equivariance constraints can be expressed using category theory. Vickers et. al provide a brief introduction to category theory for visualization practitioners [**?**], but their work focuses on data, representation, and evocation, while this paper is aims to provide guidance on how the map from data to representation should be implemented.

3 Artist

The Artist $\mathcal{A}$ is a transformation from a

We propose that the artist $\mathcal{A}$ is a function

$$A : E \to H \tag{6}$$

Definition 3.1. The category $\mathcal{K}$ consists of

1) *objects* all open sets $U_i \in \{U\}$ in K, including the empty set $\varnothing$ and the maximum set K
2) *morphisms* inclusion maps $\iota : U_i \hookrightarrow U_j$

Definition 3.2. The category consists of

1) *objects* bundles E
2) *morphisms* inclusion maps: $\iota : E_i \hookrightarrow E_j$

Definition 3.3.

3.1 Data

We model data as sections of a fiber bundle (E, π, K, F). We encode the continuity of the data as the base space K. does this go in intro? This is mostly important for data, and we kinda use it for visual intermediary but not really for H The fiber space F is the space of all possible values of the data **??**. To allow us to map multiple fields of the same datatype to distinct fibers, we adopt Spivak's formulation of the fiber as a (column name, data domain) simple schema [**?**], [**?**]. Spivak models the fiber space as a fiber bundle with sections $\sigma : C \to DT$ that take as input column name $c \in C$ and returns as output the set of values associated with the type of the field $T \in DT$, for example $\mathbb{R}$ for `floats`. In the paper, the F is the cartesian cross product of these datatype domains **DT**. We model the data as the section τbecause, as described in **??**, the section is the map from the indexing space $K$to the space of possible data values F. Spivak's notation also allows for associating elements in a section with the field it comes from. This allows for field based selection of values, while inclusion allows for continuity (index) based selections.

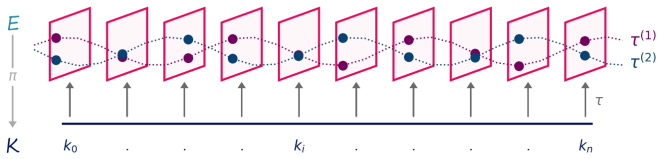


Fig. 6: replace with more concrete

One example of encoding data as a section of a fiber bundle is illustrated in **??**. In this example, the data is a not totally decided yet table of weather station data. Here we are interested in the time series of temperature values in the data, so we encode the continuity as the 1D interval K. In this multivariate data set, the fields we want to visualize are `time`, `temperature`, and `station`. The fiber space F is the cartesian cross product of the fibers of each field

$$F = F_{time} \times F_{temperature} \times F_{station}$$

where each field fiber is the set of values that are valid for the field:

$$F_{time} = \mathbb{R}$$
$$F_{temperature} = \mathbb{R}$$
$$F_{station} = \{s_0, s_1, \cdots, s_i, \cdots, s_n\}$$

The section τ is the abstraction of the data being visualized. The section at a point $k \in K$ in the base space returns a value from each field in the fiber.

$$\tau(k) = ((\text{time}, t_k); (\text{precipitation}, p_k); (\text{station name}, n_k))$$

3.2 Graphic

The object of the graphic category $\mathcal{H}$ is the fiberbundle H. The bundle H has the same structure as the data bundle E

$$\begin{array}{ccc} D & \hookrightarrow & H \\ & & \pi \downarrow \uparrow \rho \\ & & S \end{array} \tag{7}$$

with a fiber space D embedded in the total space H and a section map $\rho : S \to H$. The attributes of the graphic bundle (H, π, D, S) encode attributes of the graphic and display space

base space S continuity of display space (e.g. screen, 3D print)

fiber space D attributes of the display space (e.g a pixel = (x,y,r,g,b,a))

graphic generating function

.

We use a fiber bundle because it allows us to generalize implementation details. In this work, H assumes the the display is an idealized 2D screen, but there is no barrier to encoding for example a 3D print surface. In the same vein, ρ is an abstraction of rendering; it can be a specification such as PDF [**?**], SVG [**?**], ... or a rendering engine such as cairo or agg,

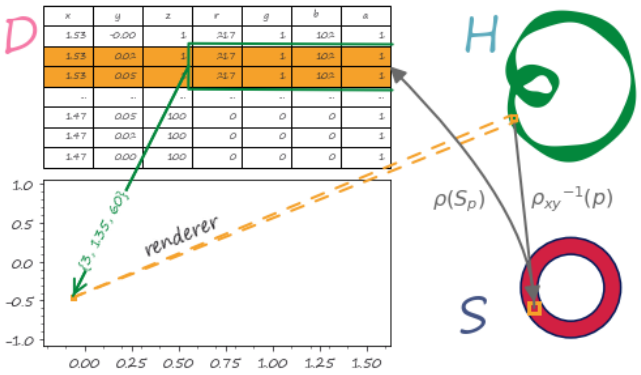
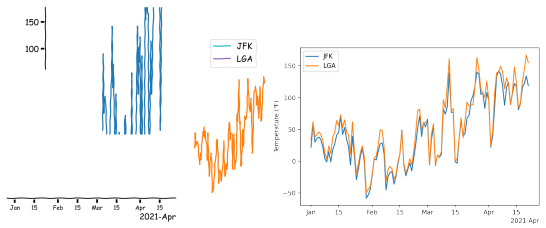


Fig. 7

Example 3.1. As illustrated in **??**,

Definition 3.4 (Category $\mathcal{H}$)**.**

3.3 Union of Artists



(a) Union of H (b) Artist with shared/harmonized μ

Fig. 8: Simulation results for the network.

4 Constraints: Construction and Formal Properties of Artists

Definition 4.1. Following from the artist Ais a special case of natural transformations

$$A : \Gamma(E, K) \nrightarrow \Gamma(H, S) \tag{8}$$

between the profunctor objects $\Gamma(E), \Gamma(H)$ that preserves continuity and equivariance.

Natural transformations are functions that map functions between categories (functors) to other functors in a structure preserving way [**?**], [**?**], [**?**].

Definition 4.2. We define the Artist natural transformation A as the tuple (ξ, ν, Q, E, V, H) where

1) ξ is a continuity preserving functor
2) ν and Q are equivariance preserving functors
3) E, V, and H are profunctor categories of sheafs (**??**)

$$\begin{array}{ccccccc} E & \xrightarrow{\nu} & V & \xleftarrow{\xi^*} & \xi^*V & \xrightarrow{Q} & H \\ & \searrow^{\pi} & \downarrow{\scriptstyle \pi} & & \downarrow{\scriptstyle \xi^*\pi} & \swarrow_{\pi} & \\ & & K & \xleftarrow{\xi} & S & & \end{array} \tag{9}$$

4.1 Graphic to Data: ξ

$$\begin{array}{ccc} E & & H \\ \pi \downarrow & & \pi \downarrow \\ K & \xleftarrow{\xi} & S \end{array} \tag{10}$$

4.2 Visual Bundle V

$$\begin{array}{ccc} P & \hookrightarrow & V \\ & & \pi \downarrow \uparrow \mu \\ & & K \end{array} \tag{11}$$

4.3 Data to Visual Encodings: ν

Visual Channel Encodings

$$\{\nu_0, \ldots, \nu_n\} : \{\tau_0, \ldots, \tau_n\} \mapsto \{\mu_0, \ldots, \mu_n\} \tag{12}$$

We enforce the equivariance constraint

$$\begin{array}{ccc} E_i & \xrightarrow{\nu_i} & V_i \\ {\scriptstyle m_r}\downarrow & & \downarrow{\scriptstyle m_\nu} \\ E_i & \xrightarrow{\nu_i} & V_i \end{array} \tag{13}$$

4.3.1 *Visual to Graphic:* Q

Visual encodings to something like marks

5 Case Study

We implement the **arrows** in **??**. `axesArtist` is a parent artist that acts as a screen. This allows for the composition described in **??**

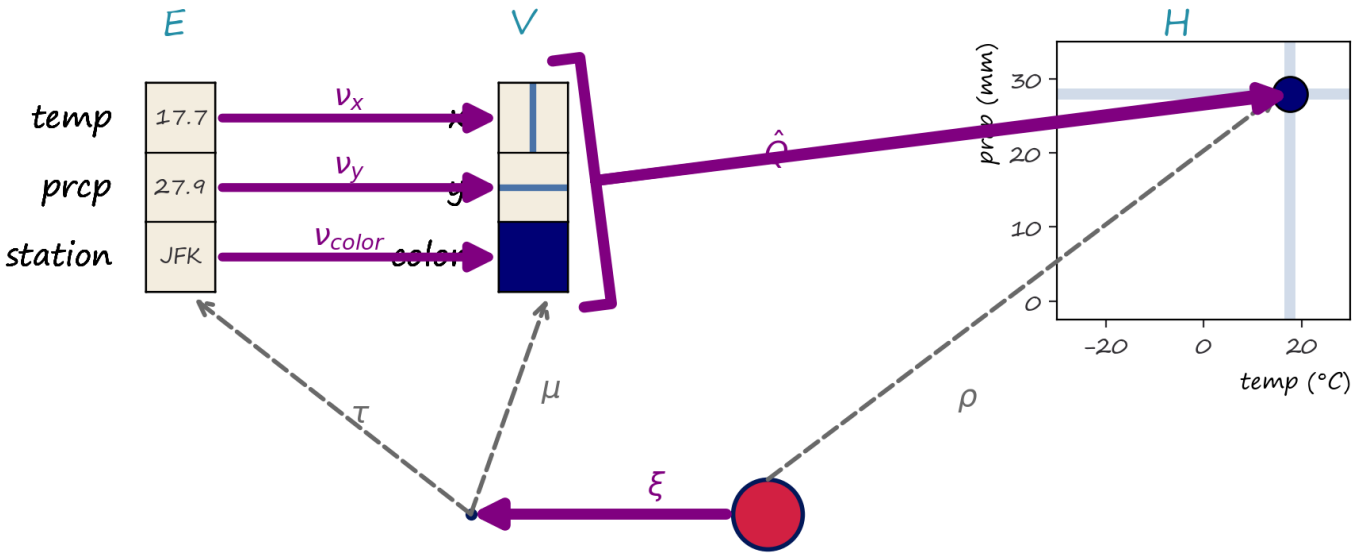
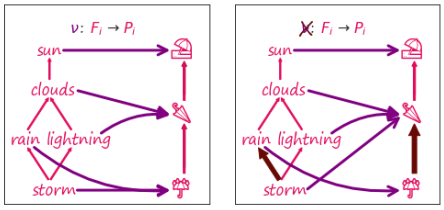


Fig. 9



(a) Artists with shared μ_i renderered correctly (b) Artists without shared μ_i

Fig. 10: Simulation results for the network.

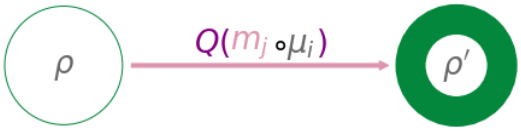


Fig. 11: rework this as a commutative box w/ the r in E row associated w/ this qhat(k)

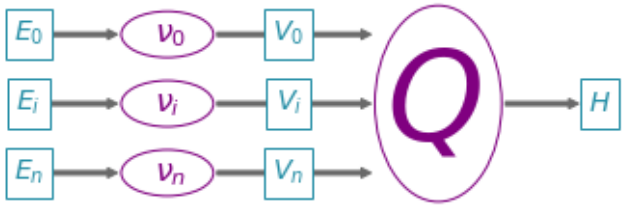


Fig. 12: add in xi!

5.1 A

```
for local_tau in axesArtist.artist.data.query(screen_bounds, dpi):
    mu = axesArtist.artist.graphic.mu(local_tau)
    rho = axesArtist.artist.graphic.qhat(**mu)
    H = rho(renderer)
```

where the artist is already parameterized with the ξ functions and which fibers they are associated to:

5.1.1 ξ

5.1.2 ν

5.1.3 $\hat{Q}$

6 Discussion

6.1 Limitations

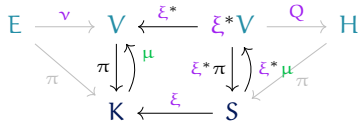
6.2 future work

7 Conclusion

The conclusion goes here.

Appendix A
Rendering: ρ

Appendix B
Manufacturing $\hat{Q} \leftarrow Q$



(14)

ACKNOWLEDGMENTS

The authors would like to thank...

Hannah Aizenman Biography text here.

Thomas Caswell Biography text here.

Michael Grossberg Biography text here.