

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 algebraic 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 suited 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**

We restrict the properties of data that should be preserved to

continuity how elements in a dataset are organized, 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

NAME	TEMP (°F)	PRCP (in.)
NEW YORK LAGUARDIA AP	61.00	0.4685
BINGHAMTON	-12.00	0.0315
NEW YORK JFK INTL AP	49.00	0.7402
ISLIP LI MACARTHUR AP	11.00	0.0709
SYRACUSE HANCOCK INTL AP	13.00	0.0118

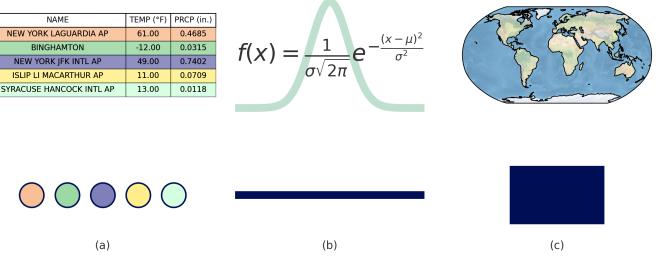


Fig. 1: Continuity is how elements in a data set are connected to each other, which is distinct from how the data is structured. The rows in (a) are discrete, therefore they have discrete continuity as illustrated by the discrete dots. The gaussian in (b) is a 1D continuous function, therefore the continuity 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 describes elements in a data set are organized; this concept is termed topological properties by Wilkinson [?]. Wilkinson provides the examples of values that are isolated from each other, and therefore discrete, and values lying on a continuum or in a compact region. 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,

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.

• 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.

58 S, SW, W). We propose that a robust model of continuity
 59 provides a way to develop library components that can
 60 work on the data in small pieces in a manner where the
 61 overall continuity of the data is preserved in the visual
 62 transformation.

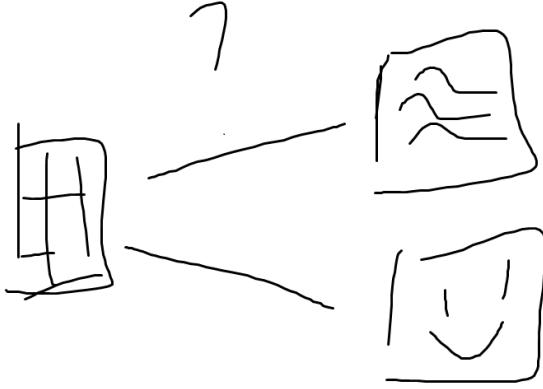


Fig. 2: Continuity is implicit in choice of visualization rather than explicitly in choice of data container. The line plots in (b) are generated by a 2D table (a). Structurally this table can be identical to the 2D matrix (a) that generates the image in (c).

The preservation of continuity can be made explicit¹⁰⁶ as in the transformation of table to parallel coordinates¹⁰⁷ in Ruchikachorn and Mueller [?], but is often expressed¹⁰⁸ implicitly in the choice of visual algorithm (visualization¹⁰⁹ type), as explored in taxonomies by Tory and Möller [?] and¹¹⁰ Chi [?].

For example, in ?? the same table can be interpreted¹¹² as a set of 1D continuous curves when visualized as¹¹³ a collection of line plots or as a 2D surface when visualized¹¹⁴ as an image. This means that often there is no way to¹¹⁵ express data continuity independent of visualization type¹¹⁶ meaning most visualization libraries will allow, for example,¹¹⁷ visualizing discrete data as a line plot or an image. General¹¹⁸ purpose visualization libraries-such as Matplotlib [?], Vtk¹¹⁹ [?], [?], and D3 [?]-carry distinct data models as part of¹²⁰ the implementation of each visual algorithm. The lack of¹²¹ unified data model means that each plot in a linked [?], [?]¹²² visualization is treated as independent, as are the transforms¹²³ converting each field in the data to a visual equivalent.

Domain specific libraries can often guarantee consistency¹²⁴ because they have a single model of the data in their¹²⁵ software design, as discussed in Heer and Agarwal [?]'s survey¹²⁶ of visualization software design patterns. For example,¹²⁷ the relational database is core to tools influenced by APT,¹²⁸ such as Tableau [?], [?], [?] and the Grammar of Graphics [?]¹²⁹ inspired ggplot [?], Vega [?] and Altair [?]. Images underpin¹³⁰ scientific visualization tools such as Napari [?] and ImageJ¹³¹ [?] and the digital humanities oriented ImagePlot [?]. macro¹³² the need to visualize and manipulate graphs has spawned¹³³ tools like Gephi [?], Graphviz [?], and Networkx [?].

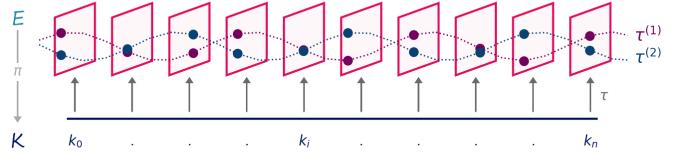


Fig. 3: A fiber bundle is mathematical construct that allows us to express the relationship between data and continuity. The **total space** E is the topological space in which the data is embedded. The **fiber** space F is embedded in E and is the set of all possible values that any **add big rectangle**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

93 2.1.1 Fiber Bundles

94 The model described in this work provides a model for
 95 expressing data sets with different topological properties.
 96 We obtain this generality by using a mathematical structure
 97 called a fiber bundles as the basis of our abstraction, as pro-
 98 posed by Butler, Bryson, and Pendley [?], [?]. As described
 99 by them, a fiber bundle is a formal model of the mapping
 100 between data points and the underlying topological space
 101 they lie in. For example, nodes on a network and the graph
 102 of the network, an image and the underlying grid at which
 103 the image is sampled, or table columns and the table index.

Formally, a fiber bundle is a mathematical structure¹⁰⁴
 (E, K, π, F)

$$F \hookrightarrow E \xrightarrow{\pi} K \quad (1)$$

Definition 2.1. The **base space** K is a topological space, which means it is a set K with a collection of open sets [?] surrounding each element in the set. Open sets are a collection of subsets $\{U\}$ in a set K , including the empty set, the whole set K , and every union and intersection of its member subsets. For each point $k \in K$, there is a collection of subsets $\{U\} \subset K$ such that $k \in U \subseteq K$ [?], [?]

intuition about base space

Definition 2.2. The **fiber space** F is a topological space such that for every $k \in K$, the fiber over that point is isomorphic to the preimage of that point $F_k \cong \pi^{-1}(\{k\})$. All fibers F_k are homeomorphic to each other [?], [?]

The fibers can be thought of as encapsulating data types according to Spivak, who proposes that databases can be represented as fiber bundles [?], [?]. Spivak formulates the fiber as a (field name, data domain) simple schema where field names and field types can be formally mapped to the set of values associated with the field type, for example \mathbb{R} for a float column named *temperature*.

Definition 2.3. The **total space** E is the space reachable via the projection map π . The total space is always locally trivial, meaning $E = K \times F$ over any open neighborhood U_k .

The fiber bundle is trivial when $E = K \times F$ is true globally; a non-trivial bundle can be thought of as the twisted fiber product $E = K \times_F F$ [?], [?]. The twisting refers to the fiber F not aligning in the same direction over all $U \subseteq K$.

As proposed by Butler et al [?], [?], data can be represented as a map from the base space K to the fiber space F .

134 This map is called a section $\tau : K \rightarrow E$ of the fiber bundle.

$$\begin{array}{ccc} F & \xhookrightarrow{\quad} & E \\ & \pi \downarrow & \uparrow \tau \in \Gamma(K, E) \\ & & K \end{array} \quad (2)$$

135 A fiber bundle has many sections, and the set of all
136 sections over an open set $U \subseteq E$

$$\Gamma(U, E) := \{\tau^i : U \rightarrow E\}_{i \in I} \quad (3)$$

137 is the set of all data sets with a shared fiber F and continuity
138 K . The set of all sections defined over the total base space is
139 denoted $\Gamma(K, E)$.

140 2.1.2 Sheaves

141 Sheaves can be thought of as an "algebraic data structure"
142 [?] for data that lives over topological spaces. A presheaf \mathcal{O}
143 is a map from the open set to the set of sections $\mathcal{O} : U \rightarrow \Gamma(U, E)$ [?], [?], [?]. The presheaf preserves inclusion maps
144 i between open sets and the inclusion map i^* between the
145 sets of sections.

$$\begin{array}{ccc} \Gamma(U_1, E) & \xleftarrow{i^*} & \Gamma(U_2, E) \\ \uparrow \mathcal{O}_E & \vdots & \uparrow \mathcal{O}_E \\ U_1 & \xleftarrow{i} & U_2 \end{array} \quad \begin{array}{c} 175 \\ (4)_{176} \end{array}$$

147 This means that a smaller space U_1 is included in a larger
148 space U_2 and a function that is continuous over a larger
149 space U_2 is continuous over a subspace $U_1 \subset U_2$. Sheaves
150 are presheaves where sets of sections are defined over unions
151 of open sets over a topology $\bigcup_{j \in I} U_j \in E$ [?], [?]. Sheaves
152 provide a mathematical abstraction of data, the space it lives
153 over, and relationships between subsets.

154 2.2 Equivariance

155 When introducing the retinal variables, Bertin informally₁₈₉
156 specifies that continuity is preserved in the mark and defines
157 equivariance constraints in terms of data and visual
158 variables being selective, associative, ordered, or quantitative
159 [?]. In the *A Presentation Tool*(APT) model, Mackinlay₁₈₃
160 embeds the continuity constraint in the choice of visualization₁₈₄
161 type and generalizes the equivariance constraint to₁₈₅
162 preserving a binary operator from one domain to another.₁₈₆
163 The algebraic model of visualization [?], proposed by Kindlmann₁₈₇
164 and Scheidegger, restricts equivariance to invertible
165 transformations.

166 2.2.1 Category Theory

167 In this work, we propose that equivariance constraints can
168 be expressed using category theory. Vickers et. al [?] provide
169 a brief introduction to category theory for visualization₁₉₃
170 practitioners, but their work focuses on reasoning about
171 visualization design, while this paper aims to provide guidance
172 on designing visualization library components. Briefly
173 we introduce concepts in category theory that we use to
174 describe our model in ?? and ??.

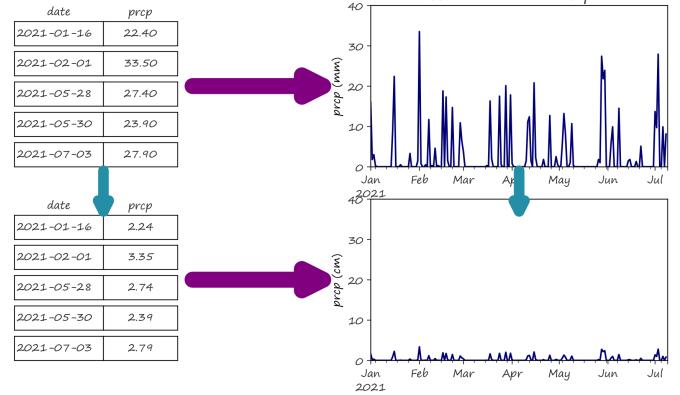


Fig. 4: Equivariance is that a transformation on the data has a corresponding transformation in the graphical representation. For example, in this figure the data is scaled by a factor 10. Equivalently the line plot is scaled by factor of 10, resulting in a shrunken line plot. Either a transformation 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.

Definition 2.4. A **category** \mathcal{C} consists of a collection of objects c with identity $\text{id}_c : C \rightarrow C$ and the set of morphisms between every two objects $f : C_1 \rightarrow C_2$, [?], [?].

The set of morphisms between objects is termed the hom set $\text{hom}_{\mathcal{C}}(C_1, C_2)$. The morphisms on the category compose

$$\begin{array}{ccccc} & & id_{C_2} & & \\ & id_{C_1} & \curvearrowright & \curvearrowright & id_{C_3} \\ & & f & \searrow & \\ & & C_1 & & C_2 \\ & & & g & \downarrow \\ & & & g \circ f & \\ & & & & C_3 \\ & & & & \curvearrowright \\ & & & & id_{C_3} \end{array}$$

and are constructed such that the following axioms hold [?]:
associativity if $f : C_1 \rightarrow C_2$, $g : C_2 \rightarrow C_3$ and $h : C_3 \rightarrow C_4$
then $h \circ (g \circ f) = (h \circ g) \circ f$
identity for every $f : C_1 \rightarrow C_2$ there exists identity morphisms $f \circ \text{id}_{C_1} = f = \text{id}_{C_2} \circ f$

Definition 2.5. An opposite category \mathcal{C}^{op} is a category with all the same objects of category \mathcal{C} and morphisms that are reversed. For example $f : C_1 \rightarrow C_2$ in \mathcal{C} is $f : C_2 \rightarrow C_1$ in \mathcal{C}^{op} .

Definition 2.6. A functor is a morphism between objects of any two categories $F : \mathcal{C} \rightarrow \mathcal{D}$. A functor has the properties of identity and composition [?]

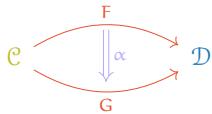
A functor preserves the structure of the category, namely morphisms and the source and target objects of those morphisms, composition of morphisms, and identity morphisms [?]. For example, the functor $F : \mathcal{C} \rightarrow \mathcal{D}$ maps an object in \mathcal{C} into an object in \mathcal{D}

$$\begin{array}{ccc} c & \xrightarrow{F} & F(c) \\ f \downarrow & & \downarrow F(f) \\ c' & \xrightarrow{F} & F(c') \end{array} \quad (5)$$

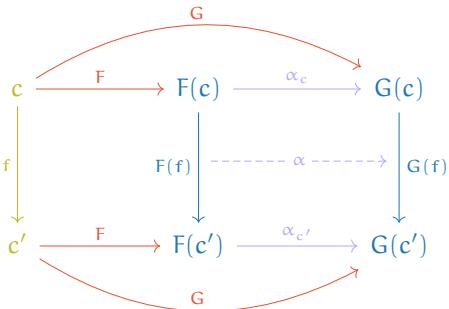
such that for every morphism $f : c \rightarrow c' \in \text{Hom}(\mathcal{C})$ there is a compatible morphism $F(f) : F(c) \rightarrow F(c') \in \text{Hom}(\mathcal{D})$.
Functors $F : \mathcal{C}^{\text{op}} \rightarrow \mathcal{D}$ are called contravariant functors.

Definition 2.7. A presheaf \mathcal{O} as introduced in ??, is a functor $\mathcal{O} : \mathcal{C}^{\text{op}} \rightarrow \text{Set}$. [?]

Definition 2.8. A natural transformation is a morphism of functors $\alpha : F \rightarrow G$ where F, G are functors from \mathcal{C} to \mathcal{D} . [?], [?]



The natural transformation composes with a functor such that the output is a compatible functor between the same categories. This means that for every morphism $f : c \rightarrow c'$ between objects in \mathcal{C} , there is a commuting naturality square for the corresponding objects in \mathcal{D} [?].



unpack this a drop more

3 ARTIST

In this section we describe the properties a data to visualization transform must satisfy to be considered equivariant and continuity preserving. We propose that by explicitly defining these properties, we can better incorporate them into visualization library design.

We formulate the visualization transformation as a functor, which we call the *Artist* A^1 . The artist A converts \mathcal{E} , which we denote \mathcal{E} to \mathcal{H} . We formulate this association as

$$A : \mathcal{E} \rightarrow \mathcal{H}$$

In this section we encapsulate the structure of the data and graphic spaces as categories. We then use these definitions to express the structure that an artist preserves. Finally we introduce rules for composing artists in a way where the structure between the inputs to the individual artists is preserved.

¹We call this transformation the artist because the *Artist* object Matplotlib [?]

3.1 Natural Transformation of Data to Graphic

Fiber bundles, as introduced in ??, serve as the model of data and graphic spaces in our model. In this section, we introduce categorical formulations of the topological spaces that make up a fiber bundle.

We introduce the category \mathcal{K} to encapsulate the subsets of the continuity and the inclusion maps that glue them together.

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

- *objects* open sets $U_i \in \{\text{openset}\}$ in the topological space (K, \mathcal{T}) , including the empty set \emptyset and the maximum set K .
- *morphisms* $\iota : U_i \hookrightarrow U_j$ for all $U_i, U_j \subset E$

We model data as the sheaf $\mathcal{O}(E) : \mathcal{K}^{\text{op}} \rightarrow \text{Set}$. The objects of Set in this sheaf are the sets of sections introduced in ?? and the morphisms are the inclusion maps in ??.

$$\begin{array}{c} \Gamma(U_i, E) \in \text{Ob}(\text{Set}) \\ \mathcal{O}(E) \xrightarrow{\varphi} \mathcal{O}(E) \\ U_i \in \text{Ob}(\mathcal{K}^{\text{op}}) \end{array} \quad (9)$$

The map $\varphi : \mathcal{O}(E) \rightarrow \mathcal{O}(E)$ is any arbitrary morphism between sheaf functors $\mathcal{O}(E)$. Following from ??, φ is a natural transform, which means all inclusions ι, ι^* are preserved as part of any φ transformation.

Should this move to down where we describe equivariance? It's not used at all currently in this section

Definition 3.2. The fiber category \mathcal{F} is a monoidal category [?], which means it is a category equipped with a bifunctor $\otimes : \mathcal{F} \times \mathcal{F} \rightarrow \mathcal{F}$

- *object* a monoid object F . A monoid is a set with a binary operator that is associative, closed, and for which the set contains an identity element i [?].
- *morphisms* $F \otimes F \rightarrow F$ and unit $i \rightarrow F$

The bifunctor \otimes combines categories in a way that preserves identity and composition [?]. For example, given a pair of categories $\mathcal{F}_\perp, \mathcal{F}_\parallel$, the bifunctor yields a set of pairs $\{(F_a, F_b) | F_a \in \mathcal{F}_\perp, F_b \in \mathcal{F}_\parallel\}$. For the associated morphisms $g : \mathcal{F}_\perp \rightarrow \mathcal{F}_\perp$ and $h : \mathcal{F}_\parallel \rightarrow \mathcal{F}_\parallel$, the function $(g \times h) : (\mathcal{F}_\perp \times \mathcal{F}_\parallel) \rightarrow (\mathcal{F}_\perp \times \mathcal{F}_\parallel)$ is pointwise $(g \times h)(F_a, F_b) := (g(F_a), h(F_b))$. Because of the bifunctor, objects of the category \mathcal{F} can be the product of any arbitrary number and types of categories, including Set, Graph , and topological spaces Top .

The set of all morphism of the fiber category $\text{Hom}_{\mathcal{F}}(F, F)$ are all possible morphisms in a given category. This includes any of the functions that serve as the basis of equivariance, such as the binary operations, group actions, and measurement scale discussed in ?? . The generalization to $\text{Hom}_{\mathcal{F}}$ also allows for the inclusion of structures such as monoid actions [?], which provide a way of applying partial order relations to data [?], such as to build multi-ranked indicators [?].

When a fiber bundle is trivial, we can dispense with localization and directly consider the fiber maps $\text{Hom}(F_k, F_k)$ over a point k as the basis of our equivariance. This is because a sheaf $\mathcal{O}(E)$ over a limit of open sets that contain

274 a point $k \in K$ is approximately the same as a sheaf over \mathbb{A}^1
 275 point k and is called the stalk $\mathcal{O}(E)|_k := \lim_{U \ni k} \Gamma(U, E)$ [?]. The
 276 fiber space over a point is embedded in the stalk over that
 277 point $F_k \hookrightarrow \mathcal{F}_k$. The sheaf maps φ restricted to the stalk are
 278 members of $\text{hom}_{\mathcal{F}}(F, F)$. The maps $\varphi \in \text{Hom}_{\mathcal{F}}(F, F)$ include
 279 the types of data transformations described in ??In ??, we
 280 assume that the fiber bundles are trivial. 307

281 The continuity of the graphic is encapsulated as a category \mathcal{S} containing open set objects U' , the complete set of
 282 opensets S that is the full space of the graphic, and the
 283 empty set. As with K , the morphisms between the open
 284 sets U' are inclusion morphisms ι . Over the graphic space
 285 $U' \subseteq S$ are a set of sections 311

$$\Gamma(U', H) := \{\rho^i : U' \rightarrow H\}_{i \in I} \quad (10)^2$$

287 which are the set of all graphics with a shared base space
 288 and fiber and are a subcategory of Set . This allows us to
 289 represent the full space of graphics as a sheaf $\mathcal{O}(H)$ 313

$$\begin{array}{ccc} \Gamma(U'_i, H) & \xleftarrow{\iota} & \{\rho^j : U'_i \rightarrow H\}_{j \in J} \\ \uparrow \mathcal{O}(H) & \nearrow A(\mathcal{O}(E)) := \mathcal{O}_A(H) & \uparrow \mathcal{O}_{\xi^*}(E) \\ U'_i & & \end{array} \quad (11)^6$$

290 The set of graphics reachable through an artist function is
 291 a subset $\{\rho^j\}_{j \in J}$ of the full set of graphics. These sets of
 292 reachable graphics over open sets U' are also objects of Set .⁷
 293 This allows us to formulate the graphic output by the artist
 294 as a sheaf \mathcal{O}_A from the graphic base space to the set of
 295 reachable graphics. 318

296 To establish a mapping between graphic and data base
 297 spaces, we introduce the surjective map $\xi : S \rightarrow K$. 320

$$\begin{array}{ccc} E & \xleftarrow{\xi^*} & \xi^* E \\ \uparrow \tau & \nearrow \tau \circ \xi & \uparrow \xi^* \tau \\ K & \xleftarrow{\xi} & S \end{array} \quad (12)^3$$

298 As shown in ??, the map between spaces ξ can be composed
 299 with section map τ such that there is a map from the graphic
 300 base space S to the fiber bundle in which the data lives E . 323

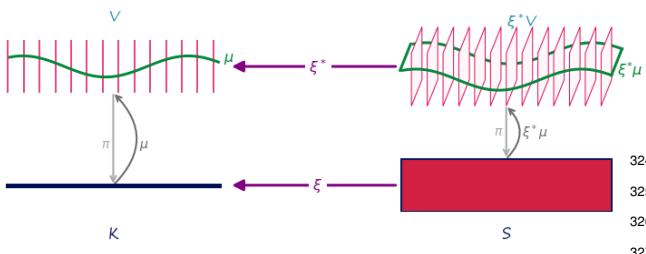


Fig. 5: The map ξ^* is a map from a repetition of the 1D fiber over every point $k \in K$ in data space, turning the fiber into a 2D plane of repeating values over the corresponding region $k = \xi(s)$ in the graphic space S . 328

Since there is a mapping from the graphic space to the total bundle $\tau \circ \xi : S \rightarrow E$, the mapping ξ can be pulled through the composition such that there is a map ξ^* between a total space over graphics $\xi^* E$ and its equivalent space over data E . As illustrated in the example given in ??, a bundle E over a point $k \in K$ is pulled back through ξ [?] such that it is copied over every point in the corresponding region in graphic space $\xi(s) = k, s \subset S$. The pullback ξ^* can then be composed with the elements of the set of sections

$$\Gamma(K, E) \xrightarrow{\xi^*} \Gamma(S, \xi^* E) \quad (13)$$

to generate a set of equivalent sections over the graphic $\Gamma(H, \xi^* E)$. As a consequence of ??, there exists a data sheaf over graphic space

$$\mathcal{O} : \mathcal{S}^{\vee} \rightarrow \Gamma(H, \xi^* E) \quad (14)$$

which is equivalent to

$$\mathcal{O}_{\xi^*} : \mathcal{S}^{\vee} \rightarrow \text{Set} \quad (15)$$

Following from ?? and the construction of the artist sheaf in ??, we propose that the artist function A is a natural transformation

$$\begin{array}{ccc} & \mathcal{O}_{\xi^*}(E) & \\ S^{\text{op}} & \xrightarrow{\quad A \quad} & \text{Set} \\ & \mathcal{O}_A(H) & \end{array} \quad (16)$$

of sheaf functors

$$A : \mathcal{O}_{\xi^*}(E) \rightarrow \mathcal{O}_A(H) \quad (17)$$

where each sheaf is a map from an open set in the graphic base space $U' \subseteq S$ into a set of sections of a fiber bundle over that open set. As defined in ??, the artist is a natural transformation. Following from ??, the artist specifies that there are compatible inclusion maps in the data and graphic spaces

$$\begin{array}{ccccc} \Gamma(U'_1, E) & \xleftarrow{\iota^*} & & \xrightarrow{\iota^*} & \Gamma(U'_2, E) \\ \uparrow \mathcal{O}_{\xi^*} & & \uparrow \mathcal{O}_{\xi^*} & & \downarrow \mathcal{O}_{\xi^*} \\ U'_1 & \xleftarrow{\iota} & U'_2 & \xrightarrow{\iota} & \\ \uparrow \mathcal{O}_A & & \uparrow \mathcal{O}_A & & \downarrow \mathcal{O}_A \\ \Gamma(U'_1, H) & \xleftarrow{\iota^*} & & \xrightarrow{\iota^*} & \Gamma(U'_2, H) \end{array} \quad (18)$$

Given that $U' \subseteq S$, ?? can be generalized to sections over S . This fact can be combined with ?? to describe the path from data sections over data space to graphic sections over graphic space

$$\Gamma(K, E) \xrightarrow{\xi^*} \Gamma(S, E) \xrightarrow{A} \Gamma(S, H) \quad (19)$$

which means that the artist, pulled back through ξ^* is a map of sections

$$A_{\xi^*} : \tau \mapsto \rho \quad (20)$$

such that for every dataset, the artist constructs a corresponding graphic generating function. Besides preserving inclusions, as described in ??, the artist function is constructed such that transformations on τ and ρ are equivariant with respect to the morphism between data sheaf functors φ which was introduced in ??.

is artist functor by virtue of being natural transform or do we define artist as functor between functors

$A : \mathcal{O}_{\xi^*} \rightarrow \mathcal{O}_A$, it follows from ?? that an arbitrary morphism φ on the data sheaf has a compatible morphism $\varphi' := A(\varphi)$ on the graphic sheaf.

$$\begin{array}{ccc} \mathcal{O}_{\xi^*} & \xrightarrow{A} & \mathcal{O}_A \\ \varphi \downarrow & & \downarrow \varphi' := A(\varphi) \\ \mathcal{O}_{\xi^*} & \xrightarrow{A} & \mathcal{O}_A \end{array} \quad (21)$$

As a consequence of ??,

3.2 Composition of Artists

Visualizations generally consist of more than one visual element, for example line plots, a legend, axis ticks, spines, and labels. We propose that we can express expected consistency between these elements as a preservation of morphisms between objects of different data categories \mathcal{E} .

3.2.1 Shared Index

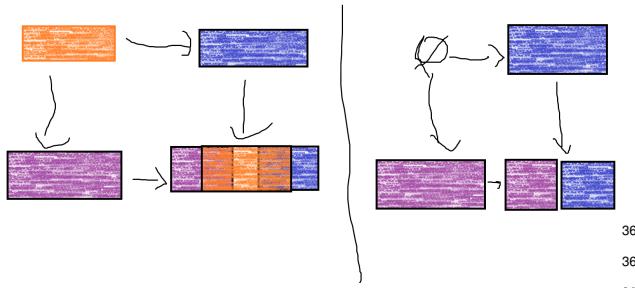


Fig. 6: In ??, the input object $\mathcal{O}_{\mathcal{E}}$ encodes a continuous function over spaces E_a, E_b and overlapping space E_c . In this figure, the visual transformation is to a point on the line on screen. An artist that preserves the continuity of the function must generate graphics such that functions evaluated $k \in K_a \cap K_b$

An example of overlapping indexing is a parallelized version of a sliding window algorithm where window overlaps must resolve to the same value at the same position ??; [?];

$$\begin{array}{ccc} K_c & \xleftarrow{i} & K_b \\ \downarrow i & & \downarrow i_{K_b} \\ K_a & \xrightarrow{i_{K_a}} & K_a \sqcup_{K_c} K_b \end{array} \quad (22)$$

where the projection functions i_{K_a}, i_{K_b} behave such that $i_{K_a}(k)|_{k \in K_b} = [k], i_{K_b}(k)|_{k \in K_b} = [k]$, meaning the projection functions yield equivalent points in the disjoint union $K_a \sqcup_{K_c} K_b$. Data that is completely disjoint in the index, as shown in ??, is a special case where $K_c = \emptyset$ and therefore there is no constraint on the artist with regards to how this data is rendered.

3.2.2 Shared Fields

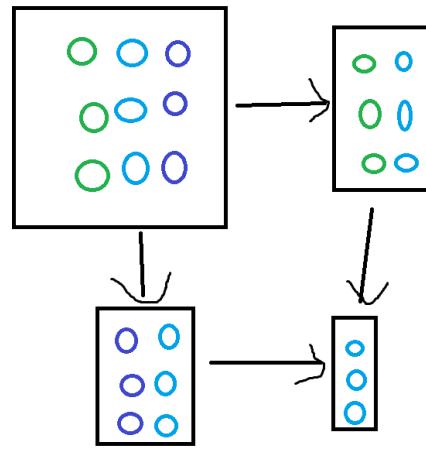


Fig. 7: The fiber spaces F_a, F_b have a shared fiber space F_c . If they are input into the same artist A_1 , then an equivariant transformation is one where F_c is transformed to a visual element in a consistent manner. In this figure, F_c is mapped to the x-axis for both F_a and F_b .

$$\begin{array}{ccc} F_a \times F_b & \xrightarrow{\text{proj}_a} & F_a \\ \downarrow \text{proj}_b & & \downarrow \text{proj}_c \\ F_b & \xrightarrow{\text{proj}_b} & F_c \end{array} \quad (23)$$

?? and ?? can be composed to specify how different aspects of the structure of a multivariate nested dimensional dataset, such as a spatio-temporal weather dataset, should be preserved in a visualization. **this is probably the power of this model and should maybe get a figure?**

4 CONSTRUCTION AND FORMAL PROPERTIES OF ARTISTS

Following from ??, the artist A is a natural transformation

$$A : \mathcal{O}_{\mathcal{E}} \quad (24)$$

that we construct to preserve continuity and equivariance.

Definition 4.1. We define the natural transformation **Artist** A as the tuple $(\xi, v, Q, \mathcal{E}, \mathcal{V}, \mathcal{H})$ where

- 1) $v : E \rightarrow V$ is a bundle map from data values to the visual variables they are mapped to
- 2) $Q : \mathcal{O}_{\mathcal{V}} \rightarrow \mathcal{O}_A$ is a sheaf map that builds a graphic generating function parameterized by the visual variables
- 3) $\xi : S \rightarrow K$ is a surjective map from the graphic topological base to the data topological base

and \mathcal{E}, \mathcal{V} , and \mathcal{H} are the categorical representations of fiber bundles that model the data, visual variable, and

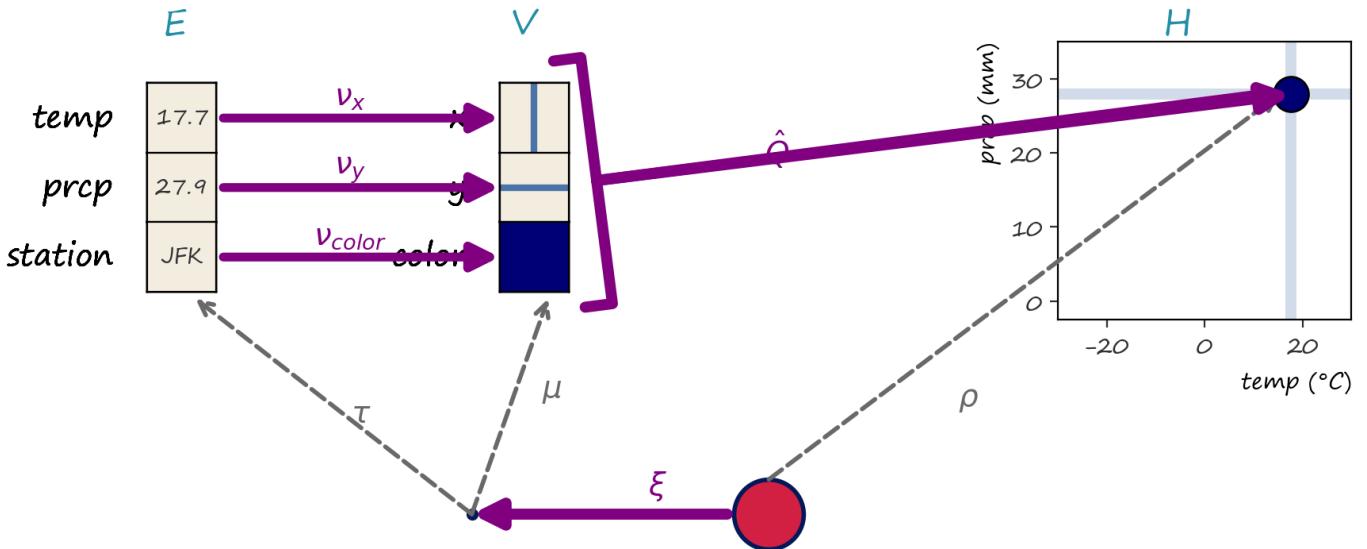


Fig. 8

graphic space of the data to visual transformation. We construct the artist to have two stages

$$\begin{array}{ccc} E & \xrightarrow{\xi} & V \\ \pi \downarrow & \swarrow \pi & \uparrow \sigma_V \\ K & & \mathcal{K} \end{array} \quad \begin{array}{ccc} V & \xrightarrow{Q} & H \\ \uparrow \sigma_V & & \uparrow \sigma_A \\ \mathcal{K} & \xrightarrow{\xi^*} & S \end{array} \quad (25)$$

which are the data to visual variable map ν at the bundle level and the visual variable to graphic map Q at the sheaf level. Usage of the bundle directly in an equation or diagram - , V, H - denotes that the function can be evaluated pointwise over $k \in K$. Use of the categorical form $- \mathcal{E}, \mathcal{V}, \mathcal{H}$ - denotes that the function is evaluated over an openset object, meaning that the function needs knowledge of both a value over a point and some information about neighboring values.

4.1 Data Domain

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 . We model the data as the section τ because, as described in ??, the section is the map from the indexing space K to the space of possible data values F .

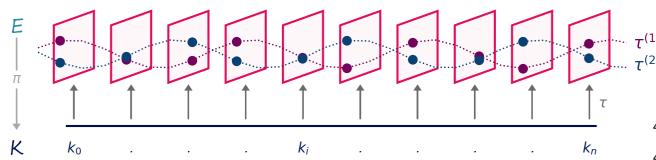


Fig. 9: replace with more concrete

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

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

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

$$\begin{aligned} F_{\text{time}} &= \mathbb{R} \\ F_{\text{temperature}} &= \mathbb{R} \\ F_{\text{station}} &= \{s_0, s_1, \dots, s_i, \dots, s_n\} \end{aligned}$$

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.

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

4.2 Graphic Codomain

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

$$\begin{array}{ccc} D & \hookrightarrow & H \\ \pi \downarrow & \nearrow \rho & \\ S & & \end{array} \quad (26)$$

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

405 **base space** S continuity of display space (e.g. screen, 3D
 406 print) 431
 407 **fiber space** D attributes of the display space (e.g a pixel 432
 408 (x,y,r,g,b,a))
 409 **section** ρ graphic generating function

410 .
 411 We represent the graphic output as a fiber bundle 433
 412 because it is an abstraction that is generalizable to various
 413 output mediums (screens, 3D prints) and types of graphics.

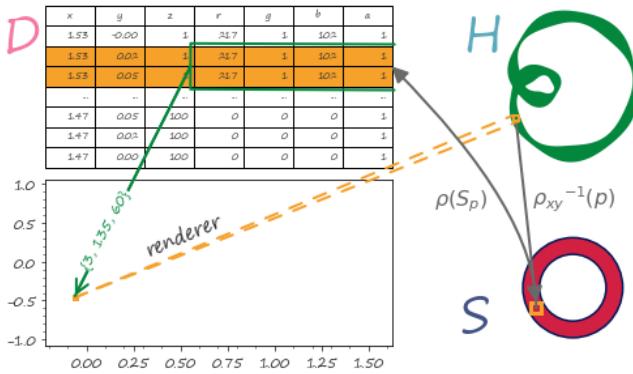


Fig. 10

414 As illustrated in ??, in this work, H assumes the the dis-
 415 play is an idealized 2D screen. include some more about the
 416 figure The graphic section ρ is an abstraction of rendering.
 417 For example, ρ can be a specification such as PDF [?], SVG
 418 [?] or an OpenGL scene graph [?], or a rendering engine
 419 such as Cairo [?] or AGG [?].

4.3 Visualization Library Components

4.3.1 Visual Bundle V

$$\begin{array}{ccc} P & \xrightarrow{\nu} & V \\ & \pi \downarrow \uparrow \mu & \\ & K & \end{array} \quad (27)$$

422 **4.3.2 Data to Visual Encodings: ν**
 423 maybe figure out how to put these side by side nicely

$$\begin{array}{ccc} E & \xrightarrow{\nu} & V \\ \pi \downarrow & \nearrow \pi & \\ K & & \end{array} \quad \begin{array}{ccc} F & \xrightarrow{\nu} & P \\ \tau \uparrow & \nearrow \mu & \\ K & & \end{array} \quad (28)$$

424 Since the functor ξ is bundle wise, it can be evaluated at
 425 the bundle at a point, which is the fiber space; therefore ξ
 426 is a functors from the product category \mathcal{F} to the product
 427 category \mathcal{P} . This constraint is expressed in our construction
 428 of ν

429 which means equivariance of

$$\begin{array}{ccc} \tau & \xrightarrow{\nu} & \mu \\ \varphi \downarrow & & \downarrow \nu_\varphi \\ \tau' & \xrightarrow{\nu} & \mu' \end{array} \quad (29)$$

Since the functor ν acts on product categories, it can be decomposed into $n u_i$ component functors that act on corresponding sections τ_i of the fiber F_i .

$$\{\nu_0, \dots, \nu_n\} : \{\tau_0, \dots, \tau_n\} \mapsto \{\mu_0, \dots, \mu_n\} \quad (30)$$

One example of this is

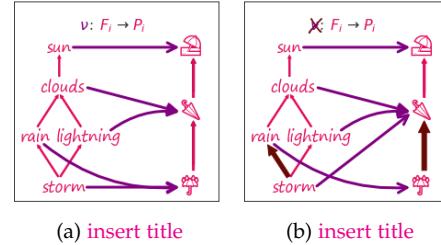


Fig. 11: is a weird nu, colors are a but much

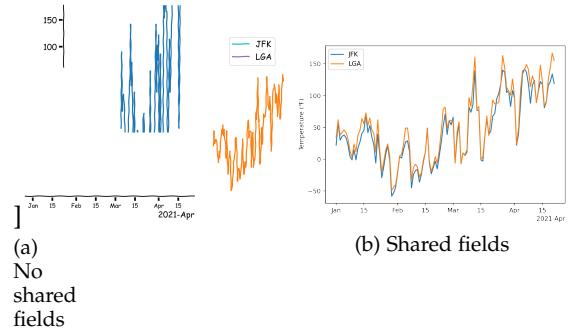


Fig. 12: In ??, the input object \mathcal{O}_E encodes a continuous function over spaces E_a , E_b and overlapping space E_c . In this figure, the visual transformation is to a point on the line on screen. An artist that preserves the continuity of the function must generate graphics such that functions evaluated $k \in K_a \cap K_b$ replace a) w/ fiber cross diagram?
 - use first half introduced in ??

434 4.3.2.1 Multiview Constraints: The concept that
 435 shared data fields should be encoded visually in a consistent
 436 is formally expressed by Qu and Hullman [?] as the notion
 437 that the same field should have the same scale.

438 We enforce the equivariance constraint specified in ?? by
 439 constructing artists that apply ν encoders such that

$$\begin{array}{ccc} F_a & \xrightarrow{\nu_c} & P_c \\ \text{proj}_a \searrow & \nearrow \nu_c & \\ F_c & \xrightarrow{\nu_c} & P_c \\ \text{proj}_b \searrow & \nearrow \nu'_c & \\ F_b & \xrightarrow{\nu'_c} & P'_c \end{array} \quad (31)$$

4.3.3 Visual to Graphic: Q

$$\begin{array}{ccc} \mathcal{V} & \xrightarrow{Q} & H \\ \mathcal{O}_V \downarrow & & \uparrow \mathcal{O}_A \\ \mathcal{K} & & S \end{array} \quad (32)$$



Fig. 13: rework this as a commutative box w/ the r in E row associated w/ this qhat(k)

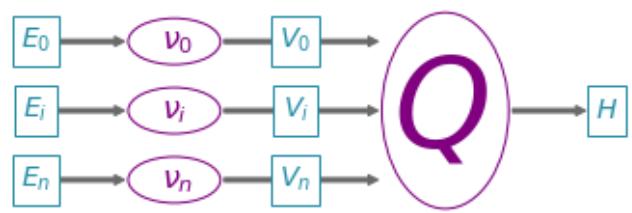
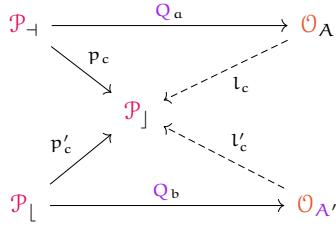


Fig. 15: v&



(33) 5.1 A

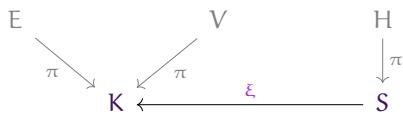
```

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

```

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

(34) 1



4.3.4 Graphic to Data: ξ

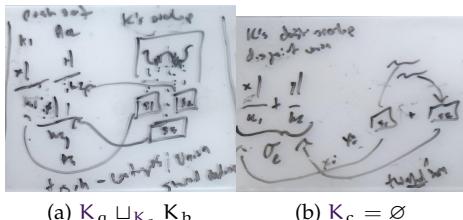
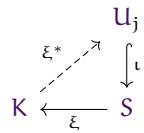
442 The functor ξ is a deformation retract, which means....

443 [?], [?]

444 By homotopy lifting?

456

457



(a) $K_a \sqcup_{K_c} K_b$ (b) $K_c = \emptyset$

Fig. 14: probably doesn't need the k=0

458 5.1.1 ξ

459 5.1.2 v

460 5.1.3 \hat{Q}

6 DISCUSSION

6.1 Limitations

6.2 future work

7 CONCLUSION

465 The conclusion goes here.

ACKNOWLEDGMENTS

466 The authors would like to thank...

468 Hannah Aizenman Biography text here.

(36)

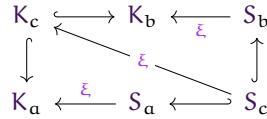
Thomas Caswell Biography text here.

Michael Grossberg Biography text here.

445 talk about xi even though is not explicitly implemented

446 The mapping between graphic and data space expresses
447 how... This is what allows the artist to generate graphics
448 where the subset of data on view is dynamically updated,
449 such as pan and zoom navigation [?] and sliding windows
450 on streaming data [?], [?].

468



5 CASE STUDY

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

470