1 Notation & Definitions

In this section we introduce a mathematical description of the visualization pipeline where artist $\mathscr{A}$ functions transform data space $\mathscr{E}$ to an intermediate representation in a prerendered graphic space $\mathscr{H}$.

$$\mathscr{A} : \mathscr{E} \to \mathscr{H} \tag{1}$$

We use fiber bundles[7, 18] to model data and graphics because they allow us to seperate the variables in a dataset from how the values are connected to each other [3, 4]. The fiber bundles mentioned in this work are assumed to be locally trivial[11, 20] unless otherwise specified.

We first describe the fiber bundle spaces of data(1.1), graphics(1.2), and intermediate visual characteristics (1.3). We then discuss the equivariant maps between data and visual characteristics (1.3.2) and visual characteristics and graphics (1.3.3) that make up the artist.

1.1 Data Space

Lets say we are working with the Global Historical Climatology Network[12], which is a dataset of global daily meterological observations. Each record includes the timestamp, location, and measurements such as temperature, precipitation, and snowfall. Using a fiber bundle allows us to express the properties of the variables(1.1.1), the connectivity of the records (1.1.3), and tables of records (1.1.4).

1.1.1 Variables: Fiber Space F

The fiber is a topological space containing the set of all records that could possibly be in the dataset, even values that do not actually exist in the dataset. For example, a fiber F that is the temperature variable from GHCN is the range of physically possible temperature values $U_{temp} = \{x \in \mathbb{R}^{+}\}$.

$$F = U_{temp} \tag{2}$$

Then lets add the time variable from GHCN $U_{time} = \{x \in (Z^{+} \mid x \geq 1011832\}$

$$F = U_{temp} \times U_{time} \tag{3}$$

such that F is the cartesian product of $F_0 = U_{temp}$ and $F_1 = U_{time}$. The elements in a given fiber set do not need to be scalers; for example the location of each record can be encoded as $U_{loc} = \{y, x \in \mathbb{R}^2 \mid -90 \leq y \leq 90, -180 \leq x - 180\}$:

$$F = U_{temp} \times U_{time} \times U_{loc} \tag{4}$$

such that one record r in the fiber would have the form $r = (\mathit{temperature}, \mathit{time}, (\mathit{latitude}, \mathit{longitude}))$. We also add the variable name and measurement scale M monoid actions to the fiber F_i such that

$$F_0 = (\mathit{temperature}, U_{temp}, M_0) \tag{5}$$

$$F_1 = (\mathit{time}, U_{time}, M_1) \tag{6}$$

$$F_2 = (\mathit{location}, U_{loc}, M_2) \tag{7}$$

or more generally (name, type, monoid action)

$$F_i = (name, U_{type}, M_i) \tag{8}$$

where type is the set of values defined by the datatype as described in Spivak's categorical formulation of databases [21]. We describe M in more detail in section 1.1.2. We can keep growing the fiber

$$F = F_0 \times \ldots \times F_i \times \ldots \times F_n \tag{9}$$

and identify the elements in the record by the variable name. We will often be working with the decoupled fiber F_i rather than the total fiber F.

1.1.2 Measurement Scales: Monoid Actions

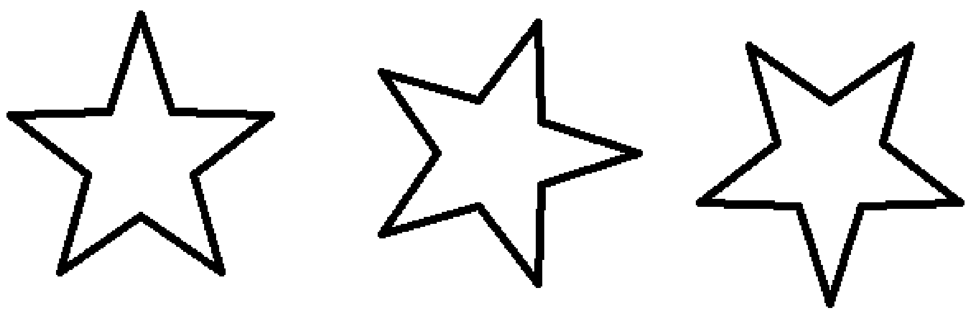


Figure 1: The set of rotation actions [turn 0, 90, 180] on the star produce a star (closed), include a rotation that does not change the orientation (identity), and can be done in any order to produce the same final orientation (closure); therefore the rotations are a set of monoid actions.

.

A common task is modifying the values in F_i in some consistent way, such as converting data from °F to °C or changing category labels or rotating a shape such as in figure **??**. This model formally encodes the set of allowable operations in the fiber because a transformation on the data side needs to be reflected in the visual representation, discussed in more detail in section 1.3.2. One method of classifying the types of operations that can be performed on data are Steven's measurment scales [10, 23]; nominal data is permutable, ordinal data is monotonic, interval data is translatable, and ratio data is scalable [24].

We generalize the set of allowable transformations to the monoid actions M_i on F_i. A monoid [13] M_i is a set with an associative binary operator $* : M_i \times M_i \to M_i$. A monoid has an identity element $e \in M_i$ such that $e * a = a * e = a$ for all $a \in M_i$. A left monoid action [1, 19] of M_i is a set U_{type} with an action $\bullet : M \times U_{type} \to U_{type}$ with the properties:

associativity for all $m, t \in M$ and $x \in U_{type}, m \bullet (t \bullet x) = (m * t) \bullet x$

identity for all $x \in U_{type}, e \in M, e \bullet x = x$

For example, the temperature F_0 defined in unit °C is on an interval scale; this means that the monoid actions M_0 on U_{temp} are the set of transformations $\{x+c \in U_{temp} \mid x, c \in U_temp\}$ where c is a valid interval between two temperatures $c = x_i - x_j$.

As with the fiber F, the total monoid space M is the cartesian product

$$M = M_0 \times \ldots \times M_i \times \ldots \times \ldots M_n \tag{10}$$

of the monoid M_i on each $F_i \in F$. Because each F_i, M_i can be independent of the other fibers, we can formally express the constraints on heterogenous multivariate datasets in the same manner as univariate datasets.

1.1.3 Connectivity: Base Space K

The fiber provides the schema of the dataset, but the fiber bundle is what glues that schema to a structure. The fiber bundle is the map π between the fiber space included in the total space E and the base space K

$$F \hookrightarrow E \xrightarrow{\pi} K \tag{11}$$

The base space K provides a way to encode how the records are connected to each other; for example the GHCN data can be expressed as discrete observations, many timeseries, maps, a network of stations, etc.

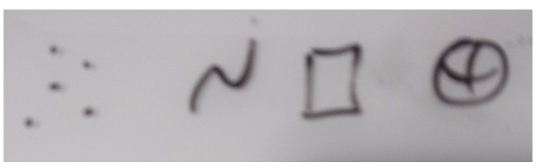


Figure 2: The topological base space K encodes the connectivity of the data space, for example if the data is independent points or a map or on a sphere

K has no knowledge of the records. As illustrated in figure **??**, K is more of an indexing space into E that describes the structure of E and itself can be any number of dimensions.

Formally K is the quotient space [16] of E, meaning it is the finest space such that every $k \in K$ has a corresponding fiber F_k in E. In figure **??**, E is a rectangle divided by vertical fibers F, so the minimal K for which there is always a mapping $\pi : F \to K$ is the line.

A fiber bundle is trivial if $E = K \times F$, which means that the fibers F are the same over every k [7, 18]. We can make many non-trivial spaces locally trivial by breaking up the total space E over local neighborhoods $U \subset K$ such that the fibers are equivalant. An example of a non-trivial fiber bundle is a mobius strip, where because of the twist some fibers are $[0, 1]$ and others are $[1, 0]$ such that we cannot construct a global map of which edge is 0 and which edge is 1. In figure **??**, the mobius strip is cut into two rectangular base spaces over $K = \{e_1, e_2\}$. We define transition functions from the fiber to the fiber along the edges of the rectangles such that we can glue these rectangles together to reform the mobius strip. A constraint we impose on the transition functions is that the monoid actions are commutative -

As with fibers and monoids, we can decompose the total space into components $\pi : E_i \to K$ where

$$\pi : E_1 \oplus \ldots \oplus E_i \oplus \ldots \oplus E_n \to K \tag{12}$$

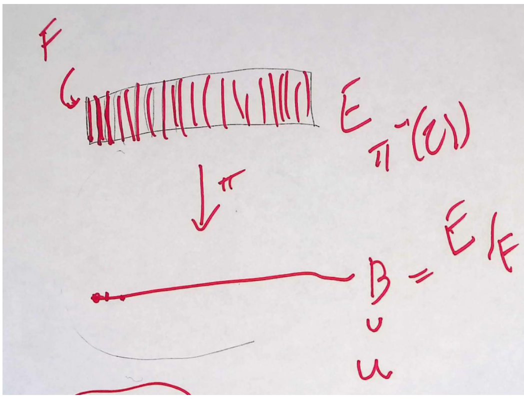


Figure 3: The base space E is divided into fiber segments F. The base space K acts as an index into the fibers.

which is a decomposition of F. The K remains the same because the connectivity of records does not change just because there are fewer elements in each record.

1.1.4 Values: Sections τ

The sections of the fiber bundle are maps from points k on K to records r in F. When we map our data to visual variables, the τ returns the set of values that are mapped together into a single graphic element. For GHCN, the τ functions return a table in a database with the schema defined in F and the connectivity defined in K: this means the sections can be defined such that all values have the same year or location, e.g each table returned by a τ is for a different station.

The section τ is the mapping from base space to total space $\tau : K \to E$

$$\begin{tikzcd} F \arrow[r, hook] & E \arrow[d, "\pi"'] \\ & K \arrow[u, bend right, "\tau"'] \end{tikzcd} \tag{13}$$

such that there is $\pi(\tau(k)) = k$ map back to K.

If the fiber bundle is locally trivial, then

$$\tau(k) = (k, (g_{F_0}(k), \ldots, g_{F_n}(k)) \tag{14}$$

which is also true when the fiber bundle is locally trivial over $U \subset K$. which we can also decompose such that

$$\tau_0(k) = (k, (x_{F_0})) \tag{15}$$

$$\vdots \tag{16}$$

$$\tau_n(k) = (k, (x_{F_n})) \tag{17}$$

which allows us to have field wise access to the data so long as there is a shared K.

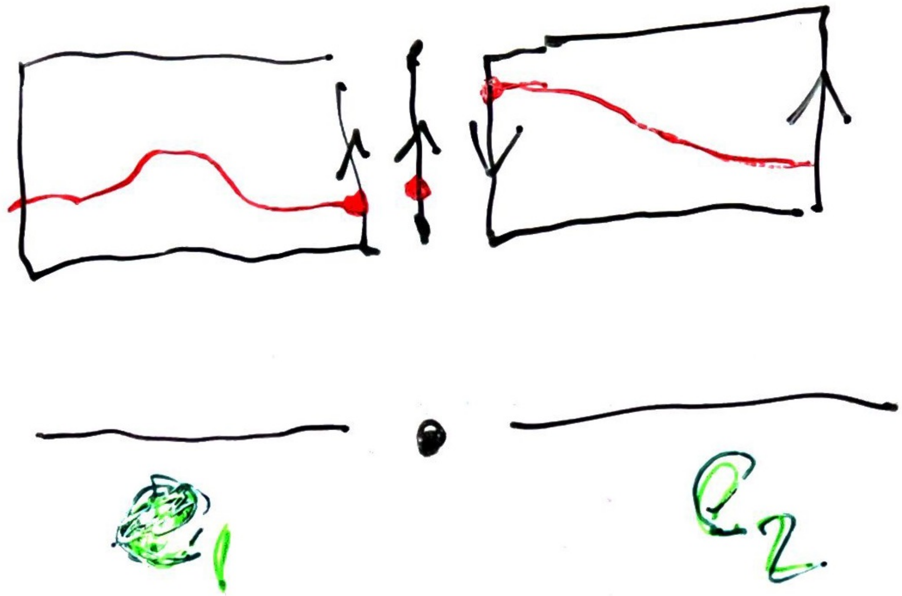


Figure 4: Many non-trivial spaces can be made locally trivial by dividing E into locally trivial subspaces and defining transition functions between the edges on K for how to glue the two subspaces such that the sigmas are continuous.

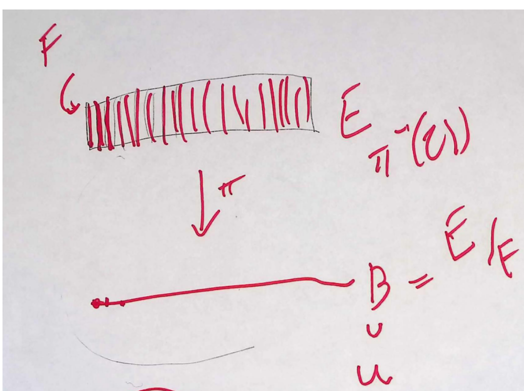


Figure 5: A section τ is a mapping from K to F that returns a record for each $k \in K$.

1.1.5 Sheaf and Stalk

Often a graphic may need to be updated with live data or support zooming in on a segement of the dataset; to support working with a subset of data, we can use the sheaf $\mathcal{O}(E)$

$$
\begin{tikzcd}
\iota^* E \arrow[d, "\pi"'] & E \arrow[l, "\iota^*"'] \arrow[d, "\pi"'] \\
U \arrow[u, bend right, "\iota^*\tau"'] & K \arrow[l, "\iota"'] \arrow[u, bend right, "\tau"']
\end{tikzcd}
\tag{18}
$$

which is the localized section of fibers $\iota^*\tau : U \to \iota^* E$ pulled back via the inclusion map $\iota : U \to K$. The localized section is the germ $\xi^*\tau$. The neighborhood of points k_i surrounding the point k the sheaf lies over is the stalk $\mathcal{F}_b$ [20, 22].

The jet bundle $\mathcal{J}$ [9, 15] is for writing differential equations on sections of fiber bundles; this information is required for some visual characteristics, such as line thickness.

1.1.6 Case Study: GHCN

Moved & walked through b/c was getting clunky to not have terms yet

The fiber bundle model is flexible enough to express some of the many different forms that temperature data can come in.

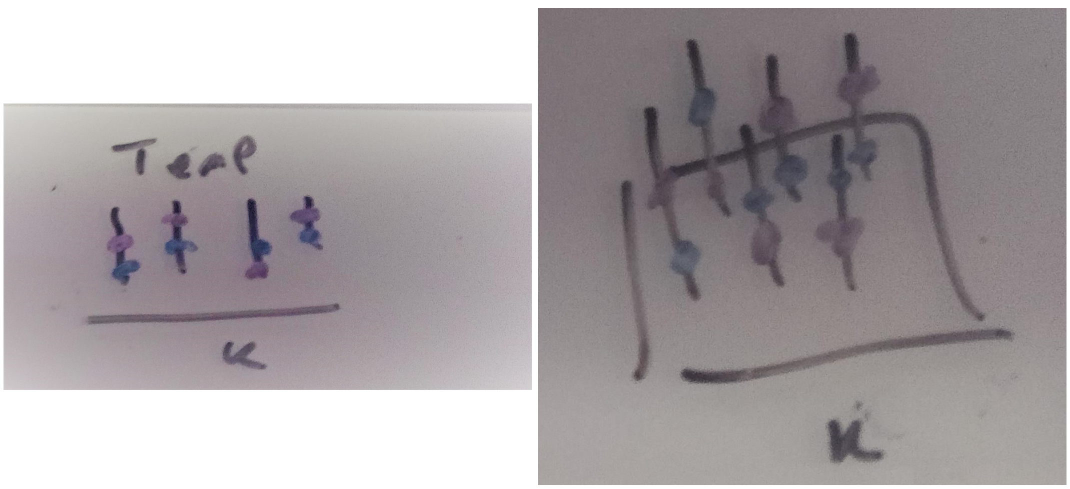


Figure 6: These two datasets have the same fiber of temperature but different base spaces. In figure **??** the temperature values are 1D continuous, while in figure **??** the temperature values are 2D continuous.

The datasets in figure **??** have identical fibers that encode a set of temperature values. In figure **??** the temperatures lie on a line such that a section could return a timeseries or a distribution. In figure **??**, the temperatures lie on a 2D continuous plane; a section could return a map or contour. Because the fiber is 1D, it does not encode metadata

To encode the metadata, the fiber is expanded as illustrated in figure **??**. The fiber in figure **??** is the cartesian product of the space of possible temperature values in degrees celsius and space of possible time values

$$F = [temp_{min}, temp_{max}] \times [time_{min}, time_{max}] \tag{19}$$

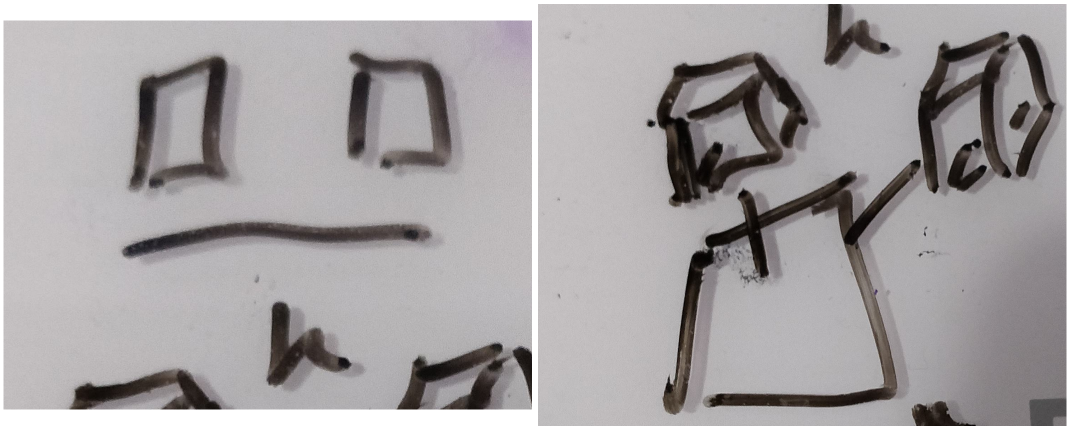


Figure 7: The fiber is expanded to include metadata fields that describe the semantics of K. In figure **??** the fiber is temperature $\times$ time and in figure **??** the fiber is temperature$\times$ latitude $\times$ longitude
.

while the fiber in figure **??** is the cartesian product of temperature, latitude, and longitude

$$F = [temp_{min}, temp_{max}] \times [-90, 90] \times [-180, 180] \tag{20}$$

such that E is the space of all possible points in F.

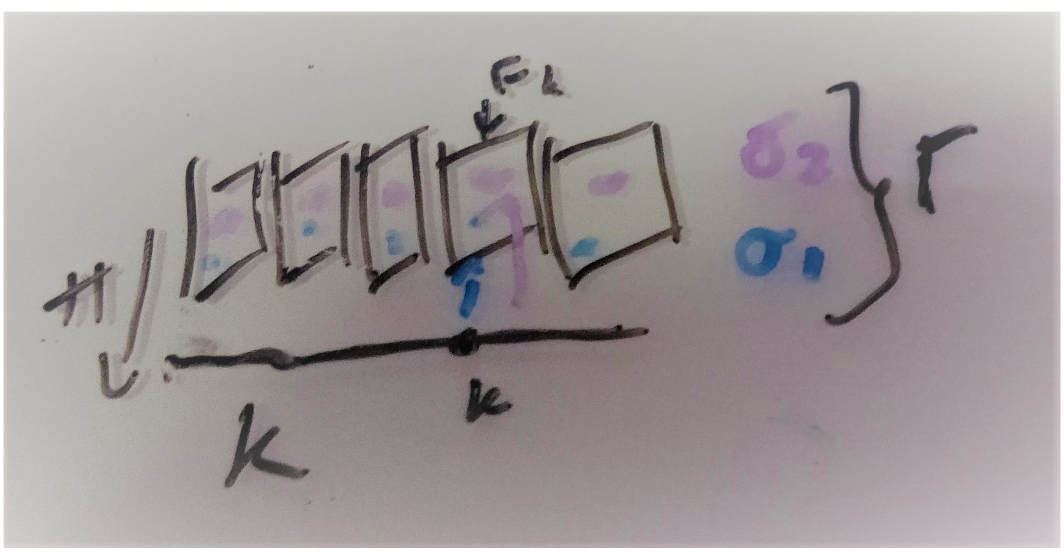


Figure 8: The section τ_1 returns the blue points, while τ_2 returns the purple points. $\Gamma(E)$ is the set of all sections, including τ_1 and τ_2
.

Given the fiber described in equation 19, the sections τ_1 and τ_2 in figure **??** return tuples of the form

$$\tau(k) = (k, (temperature, time)) \tag{21}$$

such that sections with the constraint that time is monotonic return a timeseries.

1.2 Prerender Space H

We define a graphic space H such that we do not have to assume the physical output space of the renderer. This means that the graphic in H can be output to a screen or 3D printed space or a dome. We model the prerender space as a fiber bundle (H, S, π, D). H is the predisplay space, with a fiber D dependent on the target display and a base space of S.

1.2.1 Base space

The underlying topology S of a graphic often needs more dimensions than the data topology K because of the specifications of the display space. For example, a line plot on a plane (such as a screen or a piece of paper) by necessity needs to also have a thickness so that it is visible, which maps back to a set of connected points in H. The topology of these connected points is therefore the region $s \subset S$ such that $\xi : S \to K$ is a deformation retraction [17]

$$\begin{tikzcd} E \arrow[d, "\pi"'] & H \arrow[d, "\pi"'] \\ K & S \arrow[l, "\xi"'] \end{tikzcd} \tag{22}$$

that goes from a region $s \in S_k$ to its associated point k, such that when $\xi(s) = k$, $\xi^*\tau(s) = \tau(k)$.

1.2.2 Fiber and Section

A section $\rho : S \to H$ is a mapping from a region s on a mathematical encoding of the image to a region xy on the screen that the renderer then maps to visual space as defined in D.

Example For a physical screen display, we can consider a predisplay space that is a trivial fiber bundle $H = \mathbb{R}^5 \times S$ such that ρ is

$$\rho(s) = (D_x(s), D_y(s), D_r(s), D_g(s), D_b(s)) \tag{23}$$

To draw an image, a region, H is inverse mapped into a region $s \in S$ where

$$s = \rho^{-1}_{XY}(xy) \tag{24}$$

such that the rest of the fields in $\mathbb{R}^7$ are then integrated over s to yield the remaining fields

$$R_b = \iint_s D_r(s) ds^2 \tag{25}$$

$$G_b = \iint_s D_g(s) ds^2 \tag{26}$$

$$B_b = \iint_s D_b(s) ds^2 \tag{27}$$

Here we assume a single opaque 2D image such that the z and *alpha* fields can be omitted. To support overplotting and transparency, we can consider $D = R^7$

1.2.3 Example

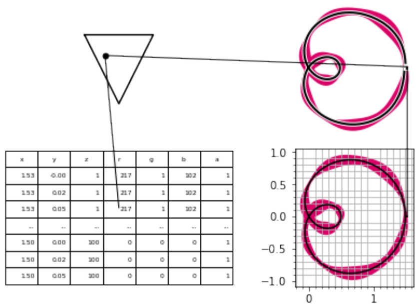


Figure 9

As illustrated in figure 9, words.

1.3 Artist

In this section we will define the artist as a mapping from a sheaf $\mathcal{O}(E)$ to $\mathcal{O}(H)$.

$$A : \mathcal{O}(E) \to \mathcal{O}(H) \tag{28}$$

The artist decomposes to mapping data to visual $\nu : E \to V$, then compositing V pulled back along ξ to ξ^*V to a visual mark in prerender space $Q : \xi^*V \to H$.

$$\begin{tikzcd} E \arrow[r, "\nu"] \arrow[dr, "\pi"'] & V \arrow[d, "\pi"] & \xi^*V \arrow[l, "\xi^*"'] \arrow[r, "Q"] \arrow[d, "\xi^*\pi"'] & H \arrow[dl, "\pi"] \\ & K & S \arrow[l, "\xi"'] & \end{tikzcd} \tag{29}$$

The pullback map ξ^* copies each value in V over k to s in corresponding S_k such that ξ^*V can have multiple values that map to one value in V.

The visual fiber bundle (V, K, π, P) has section $\mu : V \to K$ that resolves to a visual variable [2, 14] in fiber P. The visual transformer ν is a set of functions each targeting a different μ

$$\{\nu_0, \dots, \nu_n\} : \{\tau_0, \dots, \tau_n\} \mapsto \{\mu_0, \dots, \mu_n\} \tag{30}$$

where μ_i are the visual parameters in the assembly function $Q(\mu_0, \dots, \mu_n)(s) = \rho(s)$.

1.3.1 Example: Matplotlib Visual Fiber

For example, for Matplotlib [8], some of the possible types in P are:

Table **??** is an example of the visual fiber defined in terms of common parameters to plots in Matplotlib. The range of μ_i determine the monoid actions on μ_i. A section of V μ is a tuple of visual values that specifies the visual characteristics of a glyph. For example, given a fiber of $\{xpos, ypos, color\}$ one section is $\{.5, .5, (255, 20, 147)\}$. Q determines how this section is applied to a graphic.

$\boldsymbol{\nu_i}$	$\boldsymbol{\mu_i}$	$\boldsymbol{codomain(\nu_i)}$
position	x, y, z, theta, r	$\mathbb{R}$
size	linewidth, markersize	$\mathbb{R}^+$
shape	markerstyle	$\{f_0, \ldots, f_n\}$
color	color, facecolor, markerfacecolor, edgecolor	$\mathbb{R}^4$
texture	hatch	$\mathbb{N}^{10}$
	linestyle	$\{f_0, \ldots, f_n\} \times (\mathbb{R}, \mathbb{R}^{+n, n\%2=0})$

1.3.2 Visual Channels

$\nu : E \to V$ is an equivariant map such that there is a homomorphism from left monoid actions on E_i to left monoid actions on V_i where i identifies a field in the fiber. E_i and V_i each contain a set of values as defined in F and P respectively. A validly constructed ν is one where the diagram

$$\begin{tikzcd} E_i \arrow[r, "\nu_i"] \arrow[d, "m_e"'] & V_i \arrow[d, "m_v"] \\ E_i \arrow[r, "\nu_i"] & V_i \end{tikzcd} \tag{31}$$

commutes such that $\nu_i(m_e(E_i)) = m_v(\nu_i(E_i))$.

Example: Ordering To preserve ordering of elements in E_i, ν must be a monotonic function such that given $e_1, e_2 \in E_i$

$$\text{if } e_1 \leq e_2 \text{ then } \nu(e_1) \leq \nu(e_2) \tag{32}$$

Example: Translation According to Stevens, interval data is a set with general linear group actions [10, 23]. Position is a visual variable that can support translation

$$\nu(x + c) = \nu(x) + \nu(c) \tag{33}$$

Example: Invalid ν Given a transform $t(x) = x + 2$, we construct a ν that always takes data to .5:

$$\begin{tikzcd} E_1 \arrow[r, "\lambda : e \mapsto .5"] \arrow[d, "2e"'] & V_i \arrow[d, "2v"] \\ E_1 \arrow[r, "\lambda"] & V_1 \end{tikzcd} \tag{34}$$

This ν is invalid because the graph does not commute for t:

$$\nu(t(e)) \stackrel{?}{=} t(\nu(e)) \tag{35}$$

$$.5 \stackrel{?}{=} t(.5) \tag{36}$$

$$.5 \neq 2 * .5 \tag{37}$$

To construct a valid ν, the diagram must commute for all monoid actions on the sets in E_i, V_i.

1.3.3 Assembling Marks

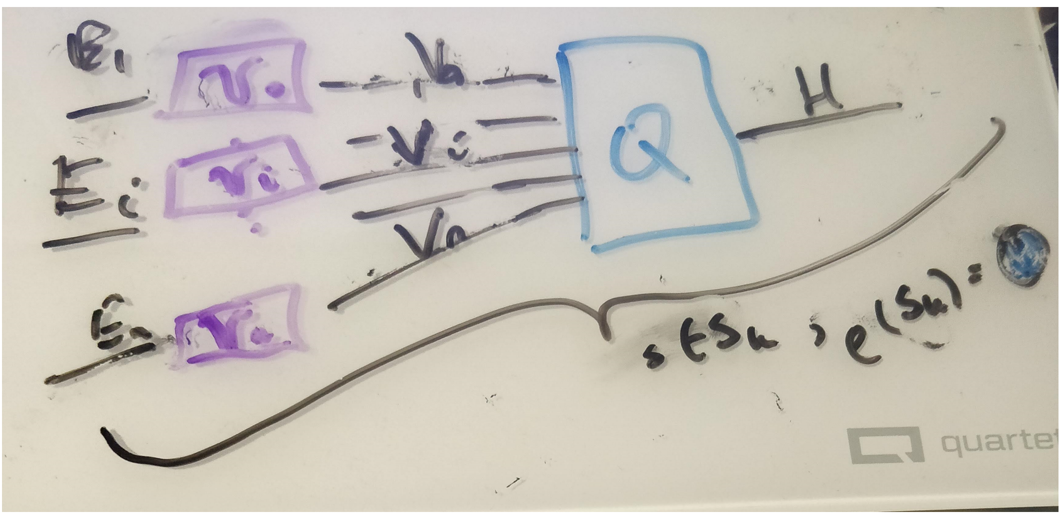


Figure 10: The ν functions convert data E to visual V. Q assembles the different types of visual parameters V_i into a graphic in H. $Q \circ \mu(\xi^{-1}J)$ forms a visual mark by applying Q to a region mapped to connected components $J \subset K$.

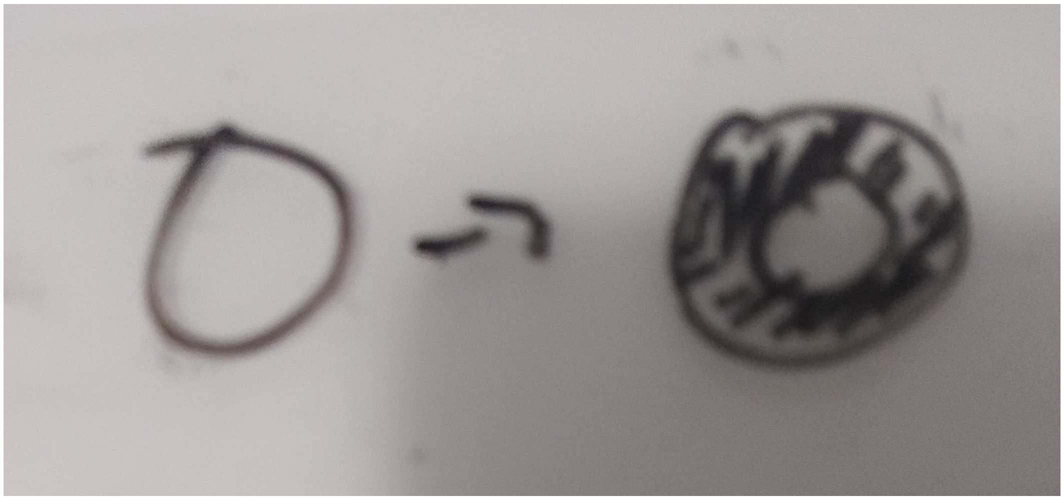
As shown in figure **??**, Q takes the individual fields in V as input and outputs a single piece of a graphic on H. As with ν, the constraint on Q is that for every monoid actions on the input in V there is a corresponding monoid action on the output in H

$$Q : \Gamma(V) \to \Gamma(H) \tag{38}$$

When $Q : \mu \mapsto \rho$ yields a ρ that maps to the same values in D over all S_k, then M can be defined over $\Gamma(H)$ such that a constraint on Q is that it must be equivariant. For example, when μ_i is the color of the glyph, it maps directly to (R,G,B) in D.

Many μ_i are graphical parameters that do not apply to the whole glyph, such as edge thickness in figure **??**.

In these situations, not all ρ in $\Gamma(H)$ will support these parameters; instead we define an action on the output graphic $Q(\Gamma(V)) \in \Gamma(H)$ since by definition every section μ will have a corresponding ρ.



We then define the constraint on Q such that if Q applied to two sections $\mu, \mu\prime$ generate the same graphic ρ, then the output of both sections acted on by the same monoid m must also be the same.

Lets call the visual encodings $\Gamma(V) = X$ and the graphic $Q(\Gamma(V)) = Y$. If $\forall m \in M$ and $\forall \mu, \mu' \in X$,

$$Q(\mu) = Q(\mu') \implies Q(m \circ \mu) = Q(m \circ \mu') \tag{39}$$

then a group action on Y can be defined as $m \circ \rho = \rho'$ where $\rho' = Q(g \circ \mu)$ with $\mu \in Q^{-1}(\rho)$.

Given

- $P = \{xpos, ypos, color, thickness\}$
- $\mu = 0, 0, 0, 1$
- $Q(\mu) = \rho$ generates a piece of the thin circle in figure **??**

the constraint on Q means that the translation $m = \{e, e, e, x + 2\}$ applied to μ such that $\mu' = \{0, 0, 0, 3\}$ has an equivalent action on ρ that causes $Q(\mu\prime)$ to be equivalent to the thicker circle in figure **??**.

Example: Invalid Q Insert some degenerate Q that generates an inconsistent glyph

Check a well defined map $M \times Y \to Y$.

constraint: inputs go to same o utput means changes to inputs mean same changes to output

Graphical Marks To output a mark [2, 5], Q is called with all the regions s that map back to a set of connected components $J \subset K$:

$$J = \{j \in K \text{ s. t. } \exists \gamma \text{ s.t. } \gamma(0) = k \text{ and } \gamma(1) = j\} \tag{40}$$

where the path[6] γ from k to j is a continous function from the interval [0,1].

We define the mark as the graphic generated by $Q(S_j)$

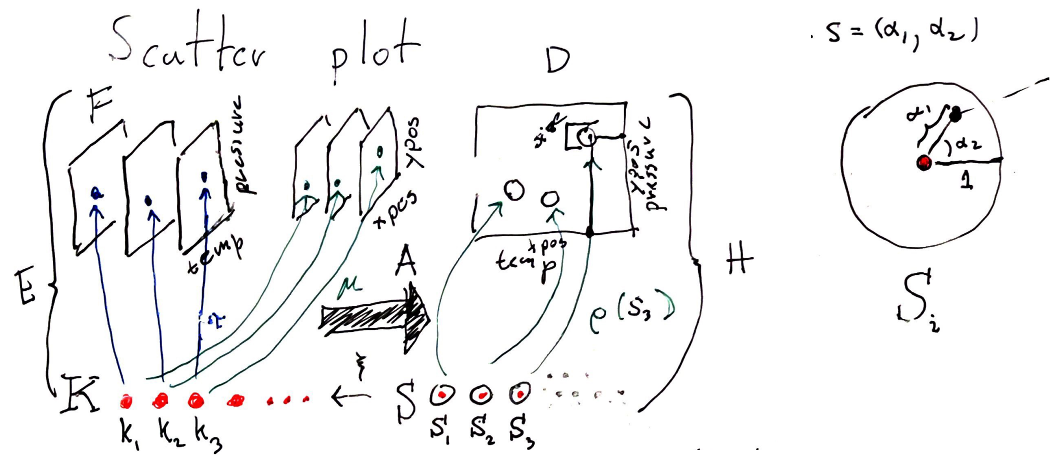
$$H \underset{\rho(S_j)}{\rightleftarrows} S_j \underset{\xi^{-1}(J)}{\overset{\xi(s)}{\rightleftarrows}} J_k \tag{41}$$

in terms of K because mark is a semantic term denoting the graphic representation of the data.

1.3.4 Sample Qs

In this section we formulate the minimal Q that will generate distinguishable graphical marks: non-overlapping scatter points, a non-infinitely thin line, and a simple heatmap.

Q: scatter plot



$$Q(xpos, ypos)(\alpha_1, \alpha_2) \tag{42}$$

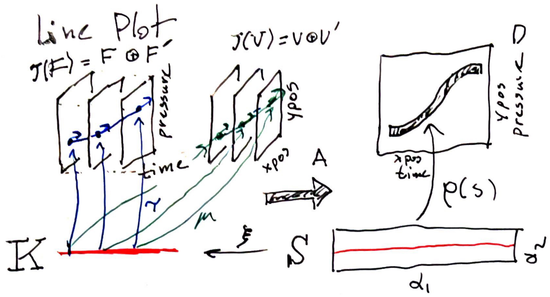
Given a defaut color of black, $\rho_{RGB} = (0, 0, 0)$. The position of this swatch of color can be computed relative to the location on the disc S_i as shown in figure **??**:

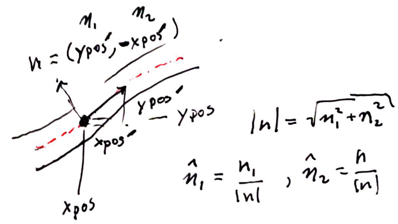
$$x = size \bullet \alpha_1 \bullet \cos(\alpha_2) + xpos \tag{43}$$
$$y = size \bullet \alpha_1 \bullet \sin(\alpha_2) + ypos \tag{44}$$

such that $\rho(s) = (x, y, 0, 0, 0)$ where s is the region in H.

Q: line plot

The line plot shown in fig **??** exemplifies the need for the jet discussed in section **??** (needs the normal to push up/down against the normal)





$$Q(xpos, \hat{n_1}, ypos, \hat{n_2})(\alpha_1, \alpha_2) \tag{45}$$

where the magnitude of the thickness is

$$|n| = \sqrt{{n_1}^2 + {n_2}^2} \tag{46}$$

such that the components are

$$\hat{n_1} = \frac{n_1}{|n|},\ \hat{n_2} = \frac{n_2}{|n|} \tag{47}$$

which yields components of $\rho(s)$:

$$x = xpos(\xi(\alpha_1)) + \alpha_2\widehat{(n_1)}(\xi(\alpha_1)) \tag{48}$$

$$y = ypos(\xi(\alpha_1)) + \alpha_2\widehat{(n_2)}(\xi(\alpha_2)) \tag{49}$$

Q: heatmap

The heatmap in figure **??**

$$Q(xpos, ypos, color) \tag{50}$$

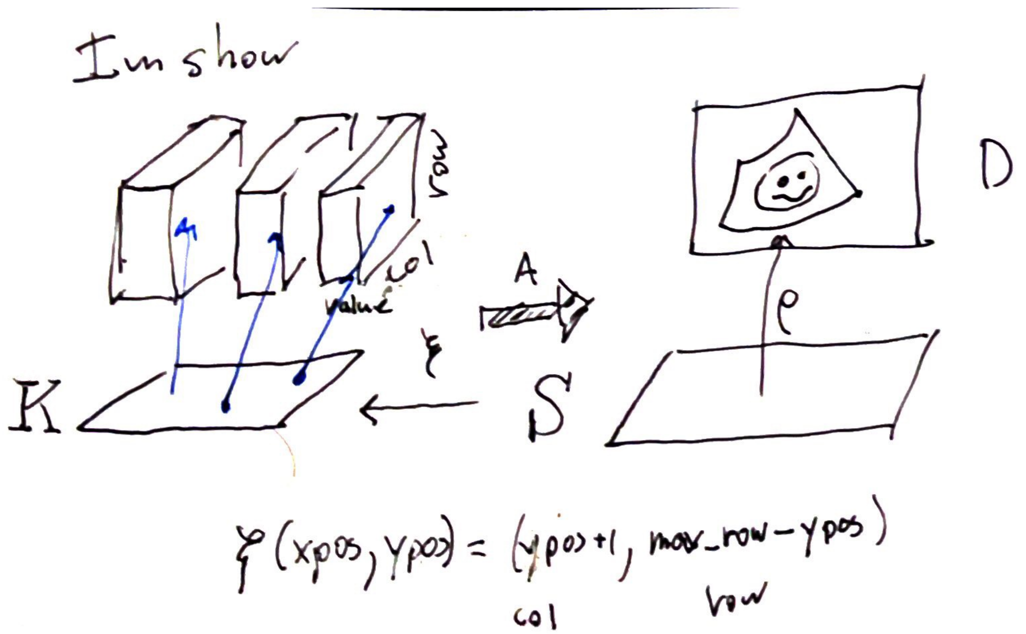
has in the simple case a direct lookup into K to obtain the $\mu = (x, y, c)$ values that are mapped into (arrays are upside down, x is y, is x) - ξ is about translating indices from data to visual alignment

$$D_{RGB} = color(\xi(\alpha_1, \alpha_2))D_x \qquad = xpos(\xi(\alpha_1, \alpha_2))D_y = ypos(\xi(\alpha_1, \alpha_2)) \tag{51}$$

through ρ.

1.4 Making the fiber bundle computable

One way to build flexible ξ is to choose a consistent way of representing K. In our draft implementation of the data as fiber bundle model, we triangularize K using complexes of the simplicies shown in figure 11 such that ξ consistently yields some combination of vertexes, edges, and faces. This gives a common data indexing structure on which to build components that could potentially be reused across Q.



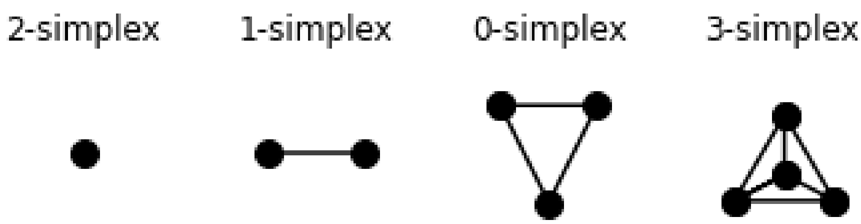


Figure 11: Simplices can encode the connectivity of the data, from fully disconnected (0 simplex) records to all records are connected to at least 3 others

By using simplicies, we can then encode the following indexing structure on τ

vertexes $\tau(k)$k= vertex id

edges $\tau(k, \alpha)$, k = edge id, α = distance along edge

faces $\tau(k, \alpha, \beta)$, k = face id, α = x on face, β = y on face

which ξ can use as part of its mapping from visual space back to data space. Path connected components are then sections where $\tau(k_i, 1) = \tau(k_j, 0)$ or the edges of the faces align.

Example Given data that lies on the toroidal space shown in figure **??**, the torus E base space K can be implemented as a simplacial complex of two 2-simplexes. We unfold

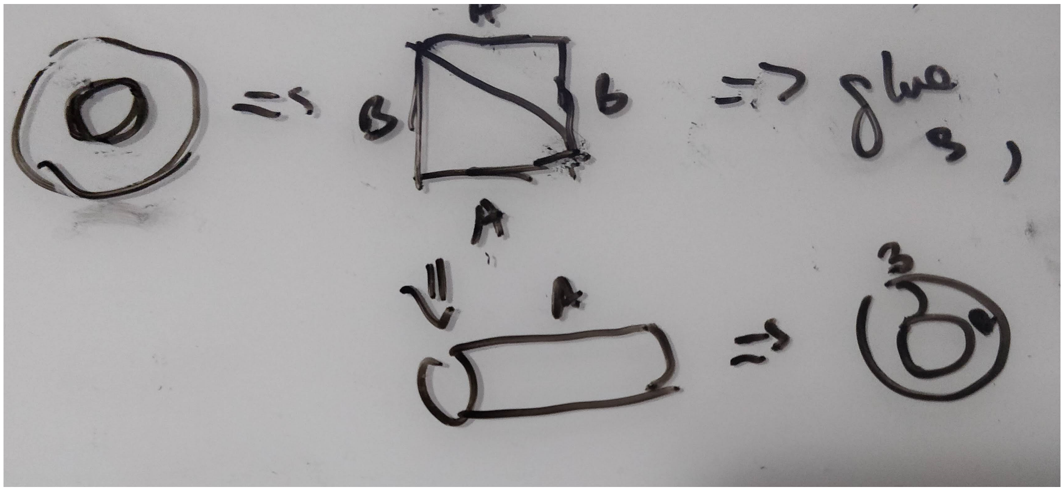


Figure 12: Representation of the base space K of a toridal total space E as two connected 2-simplices.

the torus into the two triangles that compose the square; the sections on these triangles are $\tau(triangleidk, \alpha, \beta)$. We also put transition functions on the edges such that A can be glued to A' and B to B' to reconstruct the torus.

1.4.1 Visual Idioms: Equivalance class of artists

As formulated above, every artist function A has fixed ν and generates a distinct graphic ρ. It is unfeasible to implement A for every single graphic; instead we implement the equivalence class of artists $\{A \in A' : A_1 \equiv A_2\}$ which is $Q : \Gamma(V) \to \Gamma(H)$.