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, 20] to model data and graphics because they allow us to seperate the fields in a dataset from how the values in those fields are connected to each other:

- E is a locally trivial fiber bundle over K representing data space.
- H is a fiber bundle over S representing visual space
- K and S are a triangulizable topological space or a CW complex encoding the connectivity of points in E and H respectively

The fiber bundles mentioned in this work are assumed to be locally trivial[13, 22].

1.1 Data Space

As proposed by Butler [3, 4], we model data as a fiber bundle (E, K, π, F)

$$F \hookrightarrow E \xrightarrow{\pi} K \tag{2}$$

with topological total space E, base space K, fiber space F, and the map from total space to base space $\pi : E \to K$. Maps from K to E are called sections and select specific points in K. The global space of sections in E is $\Gamma(E)$.

1.1.1 Base Space $\mathcal{K}$

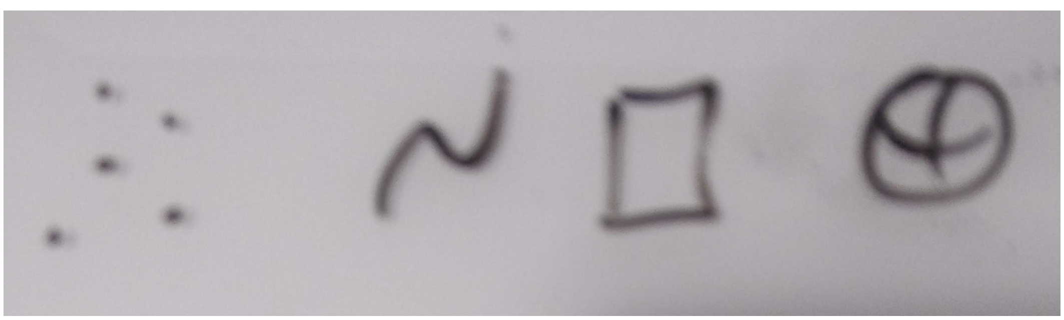
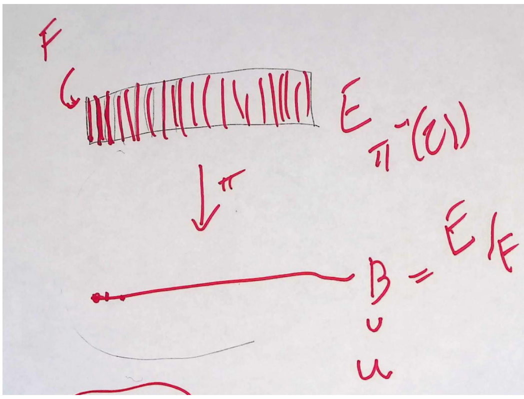


Figure 1: The topological base space K encodes the connectivity of the data space, for example if the data is independent points or a map or on a sphere

Datasets have a semantic topology such as the values are interpreted as discrete observations or part of a timeseries or map, or nodes in a networks [9, 16]. As illustrated in fig **??**, K is this underlying structure. K does not directly know the values; instead it is the sections that define the lookup between keys $k \in K$ and the corresponding values in E.



The topology K and the fields F are determined by how E is subdivided[18]. In figure **??** we can divide a rectangular base space such that there is a short fiber and long base space or a long fiber and short base space. This is analogous to long and wide forms of the same table [27].

Triangulization In our draft implementation of the data as fiber bundle model, we represent K as a simplacial complex.

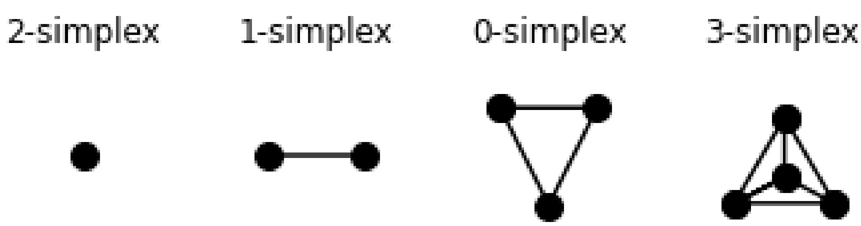


Figure 2: Simplices can encode the connectivity of the data, from fully disconnected (0 simplex) observations to all observations are connected to at least 3 other observations

K is a triangularizable topological space; one triangularization scheme is as a set composed of simplices[8], such as those shown in figure **??**.

Example Need to sketch and write this example
chopping up a torus maybe? talk about how that gets unpacked into triangles and then into vertices

1.1.2 Fiber Space

Datasets can be univariate or multivariate, homogenous or heteregenous, and fibers are where we encode this information. Each field in the dataset f_i contains a range of values that come from the codomain of f_i. The fiber is the cartesian product of the codomains of fields:

$$F = codomain(f_0) \times \ldots \times codomain(f_n) \tag{3}$$
$$F = F_0 \times \ldots \times F_n \tag{4}$$

We formulate the fiber such that it has a mapping between the codomain and the field name, as discussed in Spivak's simplacial description of databases [23, 24]. An important property of data fields are the measurement type, which we generalize to identifying the left monoid action such that the monoid actions M

$$M = M_0 \times \ldots \times M_i \tag{5}$$

is the cartesion cross product of monoids applied to each $F_i \in F$. A monoid [14] M is a set that is closed under an associative binary operator $*$ and has an identity element $e \in M$ such that $e * a = a * e = a$ for all $a \in M$.

A left monoid action [1, 21] of M is a set F with an action $\bullet$ with the properties:

closure $\bullet : M \times F \to X$,

associativity for all $m, t \in M$ and $x \in F, m \bullet (t \bullet x) = (m \bullet t) \bullet x$

identity for all $x \in F, e \in M, e \bullet x = x$

The Steven's measurement scales are identified via the group actions on F_i [12, 26] they support. For example, nominal variables are permutable and interval values are translatable. We identify monoid actions, rather than group, to support an extended variaty of scales such as partial order

1.1.3 Section

The sections of the fiber bundles are the instances of the data. They are the functions that map from a key on K to a set of values in F. For example, if we have a database, the section is the table that yields a row given an index. The section τ is the mapping from base space to total space $\tau : K \to E$

$$\begin{tikzcd} F \arrow[r, hook] & E \arrow[d, shift right, "\pi"'] \\ & K \arrow[u, shift right, bend right, "\tau"'] \end{tikzcd} \tag{6}$$

such that in a trivial fiber bundle, $E = K \times F$ [7, 20]:

$$\tau(k) = (k, (x_{F_0}, \ldots, x_{F_n})) \tag{7}$$

which we can also decompose such that

$$\tau_0(k) = (k, (x_{F_0})) \tag{8}$$
$$\vdots \tag{9}$$
$$\tau_n(k) = (k, (x_{F_n})) \tag{10}$$

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

1.1.4 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 $\mathcal{O}(E)$

$$\begin{tikzcd} \iota^* E \arrow[d, "\pi"'] & E \arrow[l, "\iota^*"'] \arrow[d, "\pi"'] \\ U \arrow[u, bend right, "\iota^*\tau"'] & K \arrow[l, "\iota"'] \arrow[u, bend right, "\tau"'] \end{tikzcd} \tag{11}$$

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 $\mathcal{F}_b$ [22, 25].

The jet bundle $\mathcal{J}$ [11, 17] is a type of sheaf that allows for writing differential equations on sections of fiber bundles; this information is required for some visual characteristics, such as line thickness.

1.1.5 Example: Temperature

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.

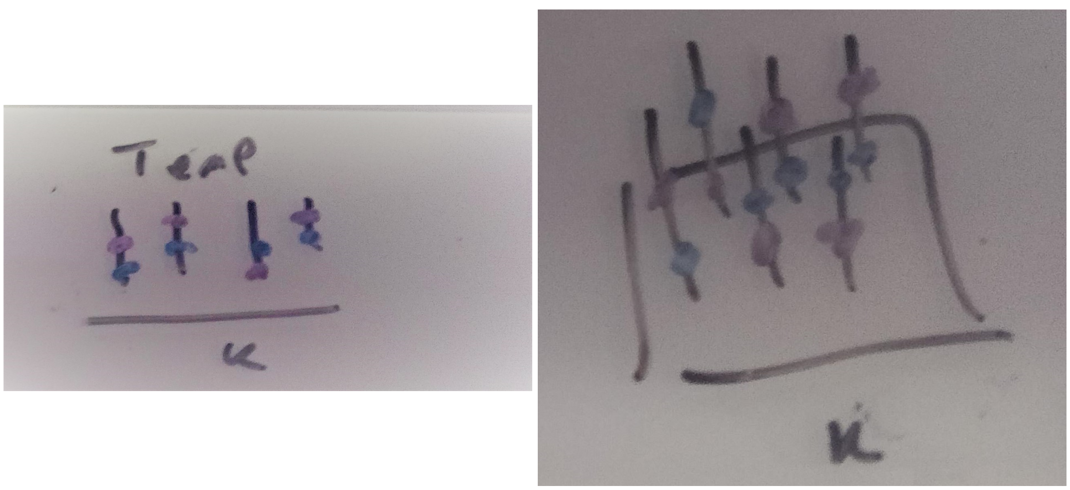


Figure 3: These two datasets have the same fiber of temperature but different base spaces. In figure **??** the temperature values are 1D continuous, while in figure **??** the temperature values are 2D continuous.

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

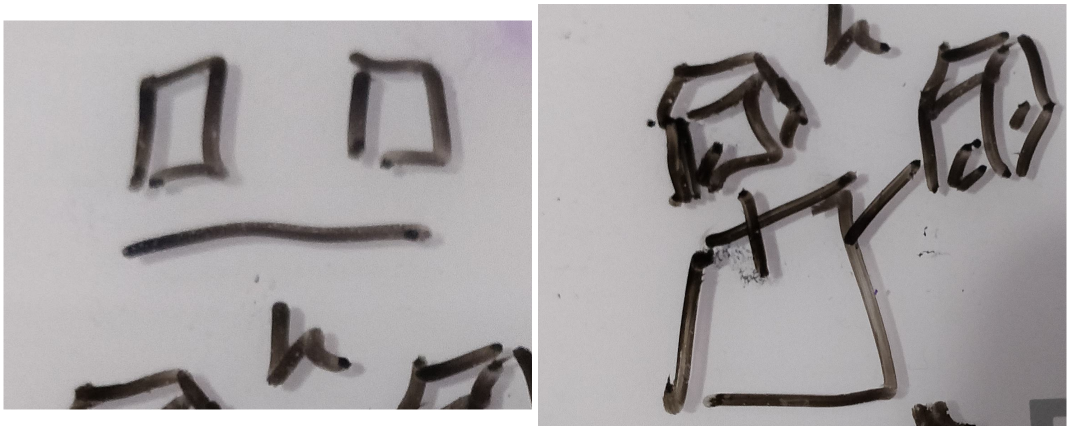


Figure 4: The fiber is expanded to include metadata fields that describe the semantics of K. In figure **??** the fiber is temperature $\times$ time and in figure **??** the fiber is temperature$\times$ latitude $\times$ longitude

.

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

$$F = [temp_{min}, temp_{max}] \times [time_{min}, time_{max}] \tag{12}$$

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

$$F = [temp_{min}, temp_{max}] \times [-90, 90] \times [-180, 180] \tag{13}$$

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

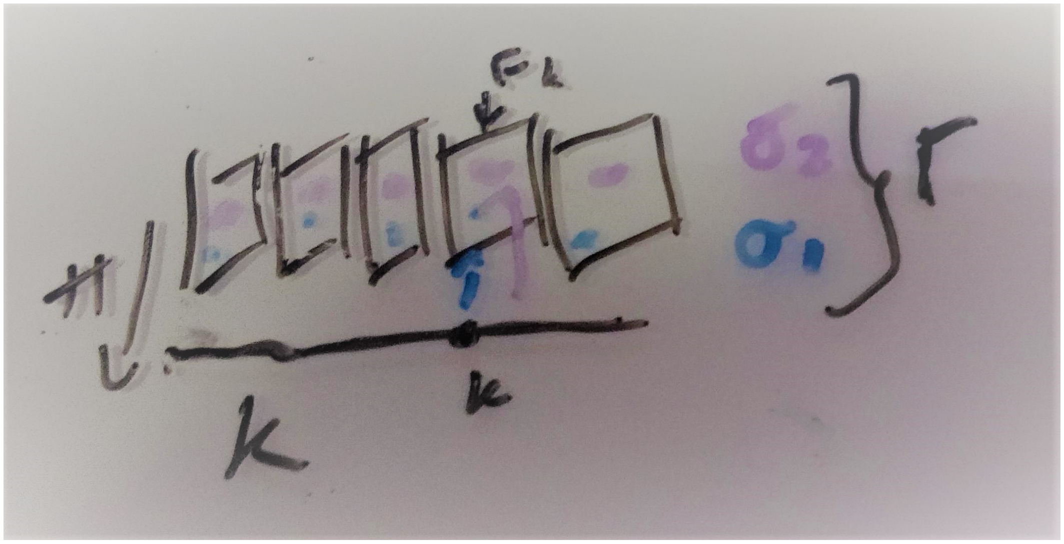


Figure 5: The section τ_1 returns the blue points, while τ_2 returns the purple points. $\Gamma(E)$ is the set of all sections, including τ_1 and τ_2

.

Given the fiber described in equation 12, the sections τ_1 and τ_2 in figure **??** return tuples of the form

$$\tau(k) = (k, (temperature, time)) \tag{14}$$

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 [19]

$$\begin{array}{ccc} E & & H \\ {\scriptstyle\pi}\downarrow & & {\scriptstyle\pi}\downarrow \\ K & \xleftarrow{\xi} & S \end{array} \tag{15}$$

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

$$\rho(s) = [x(s), y(s), r(s), g(s), b(s)] \tag{16}$$

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

$$s = \rho_{XY}^{-1}(xy) \tag{17}$$

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

$$r = \iint\limits_{s} \rho_R(s) ds^2 \tag{18}$$

$$g = \iint\limits_{s} \rho_G(s) ds^2 \tag{19}$$

$$b = \iint\limits_{s} \rho_B(s) ds^2 \tag{20}$$

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

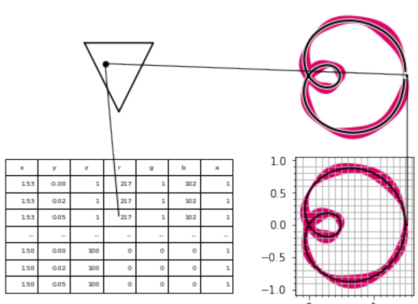


Figure 6

As illustrated in figure 6, words.

1.3 Artist

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

$$A : \mathcal{O}(E) \to \mathcal{O}(H) \tag{21}$$

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 $Q : \xi^*V \to H$.

$$\begin{tikzcd} E \arrow[r, "\nu"] \arrow[dr, "\pi"'] & V \arrow[d, "\pi"] & \xi^*V \arrow[l, "\xi^*"'] \arrow[r, "Q"] \arrow[d, "\xi^*\pi"'] & H \arrow[dl, "\pi"] \\ & K & S \arrow[l, "\xi"'] & \end{tikzcd} \tag{22}$$

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 $(V,\ K,\ \pi,\ P)$ has section $\mu : V \to K$ that resolves to a visual variable [2, 15] in fiber P. The visual transformer ν is a set of functions each targeting a different μ

$$\{\nu_0, \ldots, \nu_n\} : \{\tau_0, \ldots, \tau_n\} \mapsto \{\mu_0, \ldots, \mu_n\} \tag{23}$$

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

1.3.1 Example: Matplotlib Visual Fiber

For example, for Matplotlib [10], some of the possible types in P are:

$\boldsymbol{\nu_i}$	$\boldsymbol{\mu_i}$	$\boldsymbol{codomain(\nu_i)}$
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	$\{f_0, \ldots, f_n\} \times (\mathbb{R}, \mathbb{R}^{+^{n, n\%2=0}})$

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.

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

$$\begin{tikzcd} E_i \arrow[r, "\nu_i"] \arrow[d, "m_e"'] & V_i \arrow[d, "m_v"] \\ E_i \arrow[r, "\nu_i"] & V_i \end{tikzcd} \tag{24}$$

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 $e_1, e_2 \in E_i$

$$\text{if } e_1 \leq e_2 \text{ then } \nu(e_1) \leq \nu(e_2) \tag{25}$$

Example: Translation According to Stevens, interval data is a set with general linear group actions [12, 26]. Position is a visual variable that can support translation

$$\nu(x + c) = \nu(x) + \nu(c) \tag{26}$$

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

$$\begin{tikzcd} E_1 \arrow[r, "\lambda : e \mapsto .5"] \arrow[d, "2e"'] & V_i \arrow[d, "2v"] \\ E_1 \arrow[r, "\lambda"] & V_1 \end{tikzcd} \tag{27}$$

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

$$\nu(t(e)) \stackrel{?}{=} t(\nu(e)) \tag{28}$$

$$.5 \stackrel{?}{=} t(.5) \tag{29}$$

$$.5 \neq 2 * .5 \tag{30}$$

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

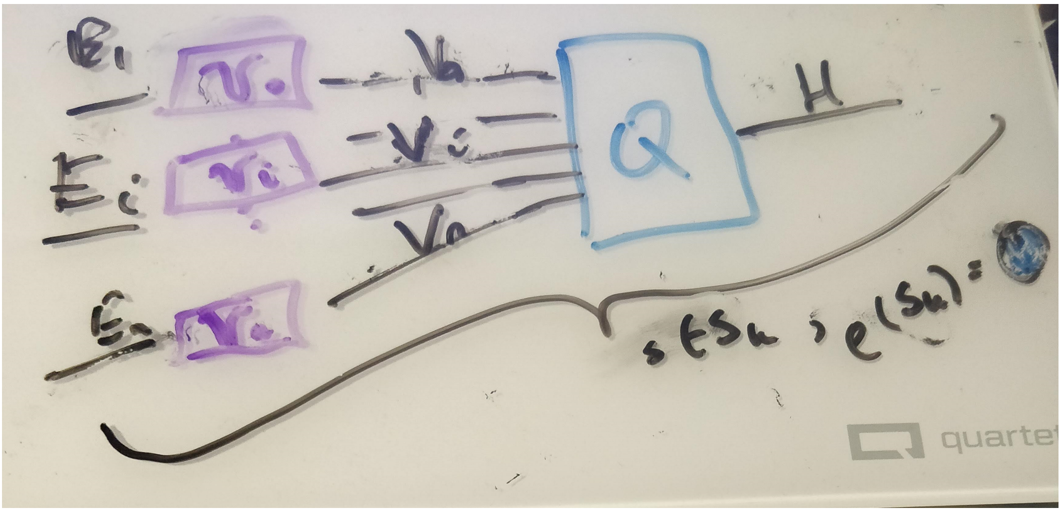
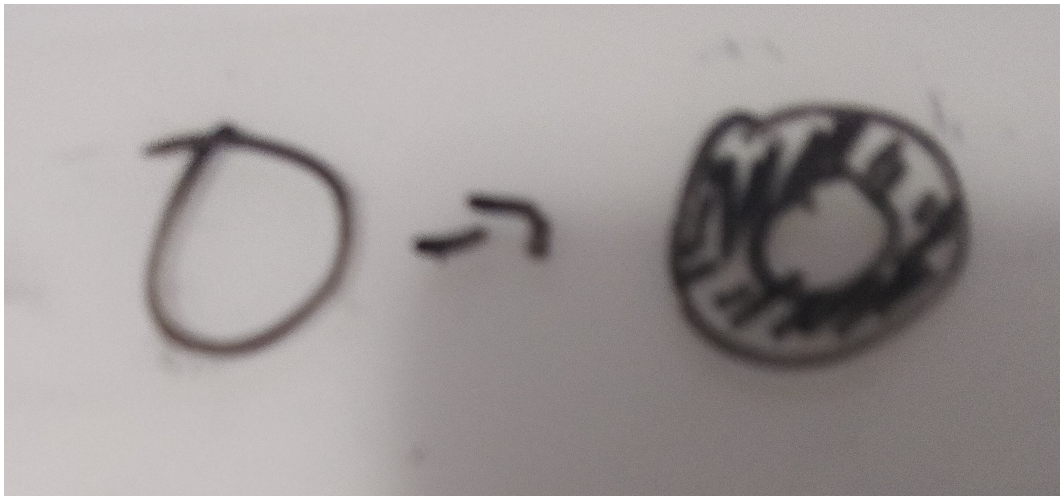


Figure 7: The ν functions convert data E to visual V. Q assembles the different types of visual parameters V_i into a graphic in H. $Q \circ \mu(\xi^{-1}J)$ forms a visual mark by applying Q to a region mapped to connected components $J \subset K$.

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

$$Q : \Gamma(V) \to \Gamma(H) \tag{31}$$

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 $\forall \mu, \mu\prime \in X$,

$$Q(\mu) = Q(\mu\prime) \implies Q(m \circ \mu) = Q(m \circ \mu\prime) \tag{32}$$

then a group action on Y can be defined as $m \circ \rho = \rho\prime$ where $\rho\prime = Q(g \circ \mu)$ 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 $\mu\prime = \{0, 0, 0, 3\}$ 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 output 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 $J \subset K$:

$$J = \{j \in K \text{ s. t. } \exists \gamma \text{ s.t. } \gamma(0) = k \text{ and } \gamma(1) = j\} \tag{33}$$

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 $Q(S_j)$

$$H \underset{\rho(S_j)}{\overset{}{\leftrightarrows}} S_j \underset{\xi^{-1}(J)}{\overset{\xi(s)}{\leftrightarrows}} J_k \tag{34}$$

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

1.3.4 Visual Idioms: Equivalance class of artists

Given $O(E)$ of the same type that output to the same type of graphic $O(H)$, the

Natural transformation + composition is partial ordering? Back and forth is equivalent