Generalizing OD-Maps to Explore Multi-Dimensional Geospatial Datasets

Liqun Liu, Romain Vuillemot, Philippe Rivière, Jeremy Boy, Aurélien Tabard

Generalizing OD-Maps to Explore Multi-Dimensional Geospatial Datasets

Liqun Liu, Romain Vuillemot, Philippe Rivière, Jeremy Boy and Aurélien Tabard

ABSTRACT
Understanding the mobility of entities in geospatial data is important to many fields, ranging from the social sciences to epidemiology, economics, or air traffic control. Visualizing such entities can be challenging as it requires preserving both their explicit properties (spatial trajectories) and their implicit properties (abstract attributes of those trajectories). An existing technique called Origin-Destinations maps preserves both explicit and implicit properties of datasets, using spatial nesting technique. In this paper we aim at generalizing this technique beyond origins and destinations dataset (2-attribute datasets), to explore multi-dimensional datasets (N-attribute datasets) with the nesting approach. We present an abstraction framework—we call GRIDIFY—and an interactive open-source tool implementing this framework using several levels of nested maps. We report on several case studies representative of the types of dimensions found in geospatial datasets (quantitative, temporal, discrete, boolean), showing the applicability of this approach to achieve visual exploratory analysis tasks in various application domains.

KEYWORDS
Exploratory Data Analysis; Geospatial Data Analysis; Grids.

1. Introduction

Entities that move or are part of geospatial datasets, such as individuals, vehicles, or traded goods, generate massive amounts of trajectory data through GPS signals, mobile phone usage, or simple postal records. Analysts rely upon such data to understand patterns of human mobility, road traffic, or trade flows. Such data usually enable the analysis of both entities (e. g., cars, countries, individuals) and their *relationships* either with themselves (if entities move over time) or across them (if entities communicate or share attribute values). Such data is often referred to as *origin-destination* (OD) data, and spatial information: they show the connections between different locations. They also contain more abstract attributes, like personal information on the individuals moving about, timestamps at which different locations are visited, or the quality and quantity of products being traded. Further attributes are usually derived, like the distance between locations, the direction of trajectories, and the time it takes to connect locations. Here, we define *explicit* and *implicit* properties to describe these two aspects of OD data. Explicit properties refer to the spatial trajectories of links between entities, and implicit properties refer to the abstract attributes of those trajectories.

Exploring OD data requires ways to rapidly navigate through implicit properties while preserving references to explicit properties. Origin Destination Map is a geospatial visualization technique proposed by (Wood et al., 2010), to encode the explicit property of the origin and destination of entities. A first level of the map encodes the origin, and a second nested level encodes the destination in cells nested on the map. Visually, this technique generates grids of maps that preserve the original map at a lower scale. This technique has been demonstrated efficient in analyzing OD data in many domains, from migration data to economic flow across

countries. Our goal in this paper is to generalize it to encode implicit properties as well as make it focus on not only OD data but also any geospatial data with nesting approach, i. e. show similar attributes between entities in the particular context of visual discovery of a dataset referred to as *exploratory data analysis*, and a tool to support this process.

Exploratory data analysis (EDA) is a highly iterative, open-ended process introduced by (Tukey, 1977). While analysts may begin with initial questions or hypotheses in mind, little is defined in advance. EDA tools are designed to help them explore data visually, so they can refine initial questions and form new ones. Combined with appropriate interaction techniques, the juxtaposition in EDA tools of even simple charts like histograms, scatterplots, or box plots—showing different facets of the data, i. e. a variety of their attributes—can prove both effective and efficient for systematically exploring large abstract datasets (as shown for example in (Elmqvist et al., 2008; Elzen & Wijk, 2014)). However, these tools are often limited when showing different implicit properties of primarily geospatial data. The most straightforward and efficient encoding proposes using circles to encode positions and connections as lines, as recommended by (Card et al., 1999). However, encoding implicit properties usually is difficult to read due to a large number of attribute values. Black and white visualizations (only) remain challenging due to the over-plot of overlapping marks; common attempts to solve this issue include using opacity to reduce the clutter of overlapping lines such as (Matejka et al., 2015), a third spatial dimension (Hurter et al., 2018), or multiple scales such as (Stolte et al., 2003; Elmqvist & Fekete, 2010). However, these techniques have their limitations, listed by (Ellis & Dix, 2007), as navigation can become challenging, according to (Moscovich et al., 2009), and the overview of the data can get lost when individual views emphasize only a few attributes.

To address these problems in an operational manner, we introduce a generalization of the OD maps techniques we call GRIDIFY and its implementation, as an interactive EDA tool for geospatial data (Figure 1). Seminal work by (Wood et al., 2010) paved the way for visualizing combinations of entities using embedded geolocation representations. Their *OD Maps* proved there are ways to leverage highly efficient visual cues like position for encoding the abstract properties of OD data without adding more clutter. Their work builds on the *hierarchical layouts* of (Slingsby et al., 2009), a technique we use in our grid-based approach. GRIDIFY enables analysts to rapidly browse all the attributes of geospatial data by progressively breaking down the dense, spatially anchored, explicit properties into subdivided facets or *cells* of a gridded interface (Figure 1 ①), according to selected implicit properties. Cells can be laid out in different ways according to various *grid patterns* (Vuillemot & Boy, 2018). GRIDIFY uses a unique, efficient encoding (the position of cells) for each new implicit property added to the visualization to not burden or distract analysts with other, more complex visual encoding mechanisms (besides picking grids). It also provides simple cues for navigating between pre-set and historical configurations. We demonstrate the expressiveness and effectiveness of GRIDIFY through several case studies, and we show how it enables the discovery of structural properties of typical datasets. We then discuss the main limitations and challenges of the generalization framework and the tool, primarily related to data preprocessing and scalability.

2. Related Work

Our focus is primarily on developing an EDA tool that relies upon OD (Origin-Destination) matrices. OD matrix is a description of movement in a certain area, with rows referring to origins and columns referring to destinations, which has been investigated by (Munizaga & Palma, 2012; D. Li et al., 2011; Ersoy et al., 2011; Iqbal et al., 2014). As we aim to generalize OD matrices visually, we investigate grid-based and faceted approaches to visualizing the trajectories of different entities in geospatial data and their abstract attributes. We consequently

Figure 1. Example of visualizations built with GRIDIFY. GRIDIFY relies on the combination of simple data grouping, aggregation, and grid patterns, to reveal implicit relationships in geo-data (e. g., correlations), while keeping explicit ones (i. e. connections) visible. ① world countries geo-trade flows (explicit relationship) as glyphes, on a heat map of total trade volume to identify top exports (implicit relationship). ② extends the same encoding as ① to nest more dimensions such as bilateral evolution by year. GRIDIFY is applicable to any geospatial dataset and we present several case studies such as analysis of football players pass ③ to reveal teams strategies.

review previous work on exploratory geovisualisation, faceted visualizations and small multiples, as well as work on constructive EDA tools developed by the Information Visualization (InfoVis) and cartography community.

2.1. Exploratory Geovisualisation

Exploratory data analysis (EDA) is a highly iterative open-ended process (Tukey, 1977). EDA tools are designed to facilitate visualization of data regardless of their statistical properties using histograms, scatterplots, or box plots. Combined with interaction, such charts can help quickly explore large datasets in a systematic manner, such as by using scatterplot matrices (Elmqvist et al., 2008), or group summaries (Elzen & Wijk, 2014).

Spatial data visualization, a rapidly expanding research area (Wood, Slingsby, & Dykes, 2011; Scheepens, Hurter, Van De Wetering, & Van Wijk, 2016), frequently employs graphs to represent spatial data entities and their interrelations through nodes and edges (Holten & Van Wijk, 2009). However, visualizing large datasets using graphs often leads to severe clutter. To mitigate this, (Holten & Van Wijk, 2009) introduced a method for bundling node-links with similar patterns, effectively reducing clutter and improving the visibility of key edge patterns. In a similar vein, (Tennekes & Chen, 2021) addressed edge clutter and occlusion by implementing named half-way points in their design. Complementing graph visualization, flow maps represent another technique for depicting flow data, using arrows or bands to illustrate 'from-to' relationships (Tobler, 1987). Demonstrating robust scalability, flow maps have been effectively utilized across various datasets (Rae, 2009). Recent trends also highlight the growing importance of human-centered approaches in flow data analysis (Dodge, Weibel, Ahearn, Buchin, & Miller, 2016), with web-based applications offering enhanced customization and design flexibility in flow data visualization (Roelandt et al., 2021). Despite these advancements, visual clutter remains a challenge in flow map visualization. To address this, researchers have developed improved flow maps using techniques such as density-based aggregation methods (Zhu, Guo, Koylu, & Chen, 2019), automated layout algorithms (Jenny et al., 2017; Sun, 2019), and even 3D visualizations (Hurter et al., 2018).

However, in the case of geo-data, their spatial nature may have led EDA tools to only support a limited range of representations, mainly 2D projections, with positions tied to circle or line marks. Graphical properties (e. g., color, width) may be used to encode an extra dimension, but it can quickly lead to important clutter because of overlaps and intersections generated from the multidimensions.

2.2. Faceted Visualizations and Small Multiples

A recurring theme in exploratory data analysis is the necessity for multiple perspectives on, or *faceted views* of given datasets—something Tufte emphasizes in his praise of small multiples as a very efficient exploration mechanism (Tufte, 1983). Mihalisin (Mihalisin et al., 1991, 1990) also introduces regular grids or lattice-like fashions to visualize the N-dimensional independent and dependent variables. Recently, Munzner (Munzner, 2014) outlines a design space for multiple faceted views, which covers the use of small juxtaposed visualizations to compare individual *perspectives* (e. g., abstract attributes) across many *collections* (e. g., small multiple views of different entities) in the data, or multiple perspectives across few collections. (Beecham et al., 2016) expand this design space by proposing a theoretical framework and an implementation of *faceted views with varying emphasis*, which attempts to tackle the issue of simultaneously visualizing multiple perspectives across many collections. Our work builds on this idea. However, where Beecham et al.'s work explores superimposing perspectives on each collection, ours breaks down the collection, e. g., the dense, spatially anchored trajectories of a geo-entity, according to the different perspectives, using a variety of grid patterns (Vuillemot & Boy, 2018). As such, our work is also related to the use of small multiples as a navigation mechanism in multivariate datasets (Elzen & Wijk, 2013) to separate subsets of the data using non-visual properties.

Google Facets[1], or the Geofacets R module[2], tie together facets and (shallow) hierarchical layouts by organizing the different views of the data in a specific spatial layout, or *grid*. However, these techniques only allow the display of a limited number of abstract attributes, since they do not support deep hierarchical nesting within facets, or *cells* of the grid. Matrix-based layouts have shown many benefits to organizing entities into rows and columns (Ghoniem et al., 2004). PivotGraph (Wattenberg, 2006) builds on 1D or 2D matrix-like grids to group nodes of multivariate graphs and let users analyze nodes properties. Pivot slices (Zhao et al., 2013), demonstrated how a faceted approach with multiple heterogeneous views could support the exploratory analysis process.

Finally, to better qualify the distinction we make between the inherent attributes of OD data and their more abstract attributes, we rely on Zhao et al.'s (2013) distinction between implicit and explicit relations in datasets. While the distinction they make is not necessarily intended for primarily geospatial data, we use the idea of explicit properties to qualify the spatially anchored aspects of geo-coded entities, and implicit properties to qualify their more abstract attributes.

2.3. Constructive EDA Tools

While the underlying motivation for facetted views is well known—to enable rapid scanning over items or attributes using descriptive visualizations–the various ways to achieve this is grounded in different strategies. Interaction is key. However, analysis attention is often split between data operations (i. e. section, queries) and visual mapping operations (e. g., graphical

[1] `https://pair-code.github.io/facets/`
[2] `https://github.com/hafen/geofacet`

encoding, choice of chart). While this offers a lot of flexibility, the two operations result in cognitive overloads and attention split. As Tufte suggest, exploratory processes should bear on *data variations*, rather than on *design variations* (Tufte, 1983). We subscribe to Tufte's view and set graphical encodings to position—the most efficient encoding (Cleveland & McGill, 1984). Our approach relies on a top-down, gridded approach to creating facets: the display space is first divided according to the values of a first, specified implicit property (e. g., time or any categorical values) into a gird of cells, each showing a relevant subset of the explicit property, and this procedure is then repeated (up to) as many times as there are implicit properties in the data.

This progressive construction of the visual display alongside the EDA process has the potential to facilitate sensemaking activities (Park et al., 2017). Using a code-driven approach, construction can build upon rich visualizations grammars, like Vega (Satyanarayan et al., 2014), Grammar of Graphics (Wilkinson, 2005), or ATOM (Park et al., 2017). These toolkits enable the creation of sophisticated visualizations, but they can be tedious to specify, and they provide limited feedback. However, these code-driven specifications also lend themselves to constructive interfaces. Tableau, building on Polaris (Stolte et al., 2008), maybe the most prominent one. These tools enable analysts to rapidly create and explore multi-dimensional data. More recently, Voyager (Wongsuphasawat et al., 2016), building on the Vega-lite grammar (Satyanarayan et al., 2017), has provided a demonstration of how construction mechanisms can be used to create multiple views and can be augmented with automated design recommendations.

Research in Computer-Aided Design (CAD) tools highlighted long ago that short feedback loops are instrumental in supporting exploratory processes. Supporting a quick assessment of alternative choices (Terry & Mynatt, 2005) can narrow down Norman's gulf of execution (Norman, 2002), and foster reflection in action (Schon, 1984). Strong signifiers (or perceived affordances), suggested interactivity (Boy et al., 2016), and feedforward (Vermeulen et al., 2013) can further support exploration and sense-making. Scented widgets (Willett et al., 2007) that display data distributions on top of widgets are a great example of *static* feedforward. A similar example is interactive legends (Riche et al., 2010). Tools like ScatterDice (Elmqvist et al., 2008), extended this principle by offering multiple views that give a hint as to what the next step of the exploration process could be. But this comes as the cost of precious screen space. *Sequential* feedforward strategies (Vermeulen et al., 2013) are dynamic, and provide immediate continuous feedback when exploring alternative choices. Continuous interactions, such as brushing techniques, are frequently utilized in various applications, such as the Octopocus command selection technique (Bau & Mackay, 2008) and Terry's generative design selection tool (Terry & Mynatt, 2005), as mentioned by (Q. Li et al., 2003). Paradoxically, creating tighter feedback loops enables users to explore in more breadth the parameter space and spend less time assessing every design choice (Terry & Mynatt, 2005).

GRIDIFY assumes a constructive approach to the creation of gridded, faceted views of geospatial data, by facilitating the creation of hierarchical layouts of subsets of their explicit properties, progressively nested according to selected properties.

3. GRIDIFY Framework: Decoupling Data and Visual Abstractions

We develop a flexible approach to address exploratory data analysis (EDA) questions for general geospatial data by decoupling the data transformation and visualization construction operations based on OD maps generalization. This allows us to more effectively handle various data types and visualize them in ways that better meet the needs of users. As illustrated in Figure 2, the InfoVis pipeline generally implies following a data-to-display analytic approach (Vuillemot & Boy, 2018). We argue that exploratory processes can benefit from going back-and-forth

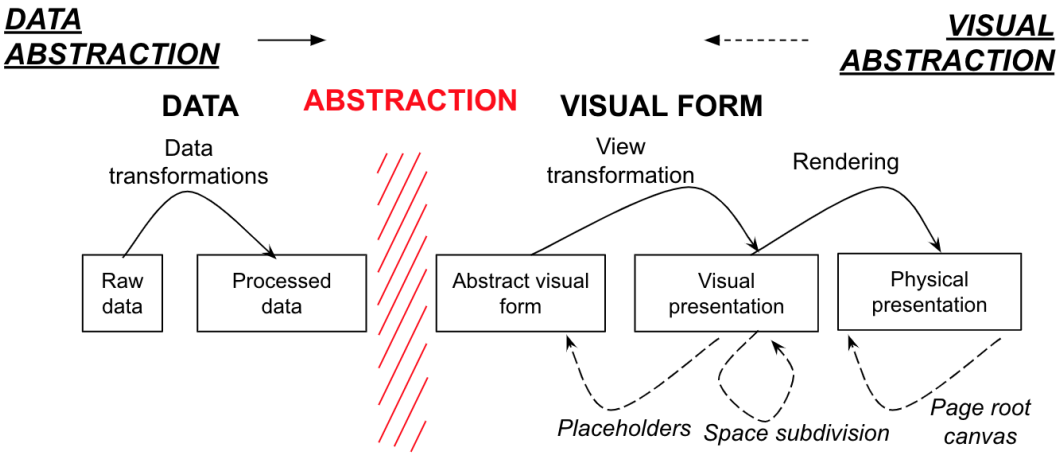

Figure 2. Graphical representation of the classic InfoVis pipeline (Card et al., 1999). Our GRIDIFY technique operates at the interface of *processed data* and *abstract visual forms* represented using red lines.

between display-to-data *and* data-to-display approaches to refine and answer sophisticated questions (Slingsby et al., 2009); and that ideas on how to best transform the data to answer given questions can be formed at a purely visual level, by manipulating and progressively constructing the elements of the display. Essentially, this means that analysts should be able to break down the explicit properties in OD data either by grouping subsets of the explicit properties in data space according to specified implicit properties, or by isolating aspects of the explicit properties directly in the visual space.

We consider the Infovis pipeline as the combination of two abstraction levels: a *data abstraction* level, and a *visual abstraction* level. We use *data abstraction* as an umbrella term for all the manipulations and transformations an analyst can perform in the data space—typically to transform large datasets into (often smaller) sets with derived attributes—before trying to map them to the graphical space (as shown in Figure 3); and *visual abstraction* also as an umbrella term for all the manipulations and constructions an analyst can perform in the graphical space, before trying infer the necessary transformations in the data space. In this section, we detail the specific data and visual abstractions we propose in GRIDIFY, and we discuss how they can be joined through nesting operations.

3.1. Data Abstractions

We consider each geo-entity as the triplet $E = \langle d, t, A \rangle$ with $d_i = (x_i, y_i)$ their explicit relation (a position in a 2D space S), ordered over time $t_1 < t_i < t_n, t \in T$ a time period, and a set of implicit properties A. Connections are two entities connected with each others $C = \langle (E_o, E_d), A' \rangle$, in most case they represent the endpoints of a trajectory, which also contain implicit properties A' (e. g., distance between entities). Those two sets of implicit property, A and A', are our focus of attention as they are responsible for the main visual abstractions in GRIDIFY, as shown in Figure 3. A and A' are composed of:

- **A dimensions list** that consists of all the implicit properties available for entities and connections $D = \{A_1, A_2, .., A_n\}, A_i \in \{A, A'\}$. Some dimensions are available permanently (*explicit dimensions* like position, time), while others are only available during, or after given data transformations (*dynamic dimensions* like speed or acceleration).
- **Grouping methods** that group values, for a dimension, based on criteria (e. g., number of expected groups) or a property of the dimension: *Ex: grouped by values (categories),*

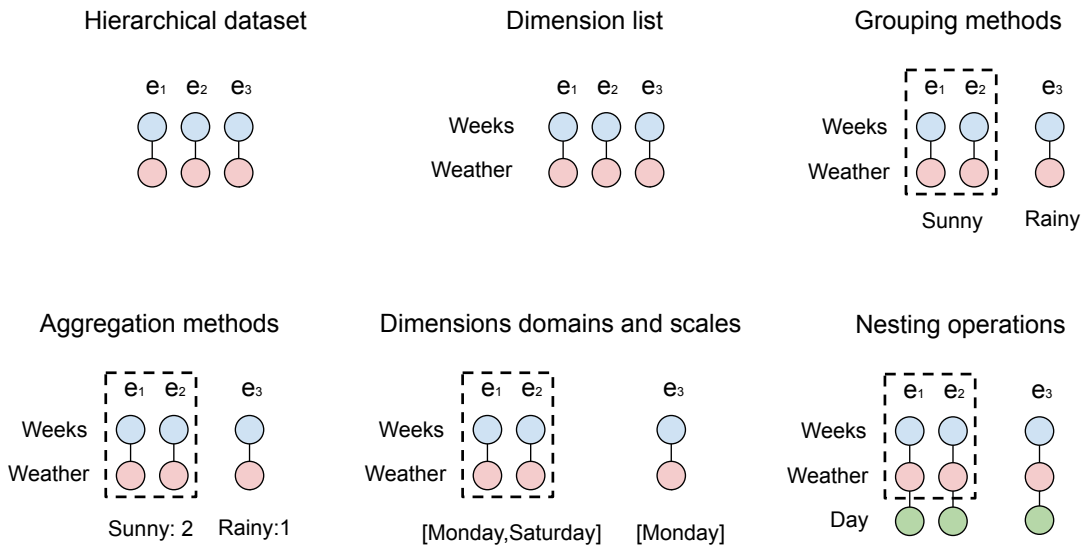

Figure 3. Data abstraction in GRIDIFY. Dimension list consists of two implicit relations (Weeks and Weather). Grouping the entities in spatial data (e. g., e_1, e_2, and e_3) based on the *Weather* dimension achieves two groups. One group contains e_1 and e_2 that only has *Sunny* value. In contrast, another group contains e_3 that only has *Rainy* value. Aggregation method estimates the statistic of grouped groups (e. g., one group has two elements and another has one element). Dimension domain method shows the value scales from other dimensions (e. g., one group has a domain in $[Monday, Saturday]$). Nesting operations generate another level of nesting (e. g., adding the *Day* dimension).

bins or buckets (quantities), sampling, etc.

- **Aggregation methods** that derive one or multiple values from the grouped group: *Ex: count, sum, mean, median, average, distinct, min/max, unique, value, etc.*
- **Dimensions domains and scales** that return a list of all unique values for a given implicit property A: $\{a1, a2, ..., am\}, a_i \in Dom(A_i), i \in [1, m]\}$ or the extent of a scale $[Min(a_i), Max(a_i)], i \in [1, m]$ for a quantitative dimension. *Ex: Dom(Years) = 2011, 2012, etc.*
- **Nesting operations** that consecutively pass down operations from one level to the next one. *Ex: Select a* `Country` $\bowtie$ *Divide by* `Year` $\bowtie$ *sum(*`Exports`*)*

Grouping and aggregation methods provide an array of options (Furmanova et al., 2017; Amar et al., 2005; Vuillemot & Boy, 2018) well documented in e. g., (Stolte et al., 2003), and nesting operations are central, as they enable increasing the number of implicit properties A and A' in the visualization.

3.2. Visual Abstractions

The main visual abstractions in GRIDIFY are space partitions we call *cells*, which are organized according to *grid patterns*. There are as many cells in a grid as there are groups in a nested data partition. As such, cells and grids also follow a nested, hierarchical structure, starting at the canvas (or *root*) level, and progressively breaking down each cell into smaller partitions (or *leaves*). We use the following notation (originally proposed in (Vuillemot & Boy, 2018)) to describe grid properties:

- **Grid patterns:** the methods define the space division method that will create **cells** which are a space partition with coordinates and dimensions, as shown in Figure 4 (a).
  Parameters: grid, horizontal, vertical, coordinates, treemap, central,

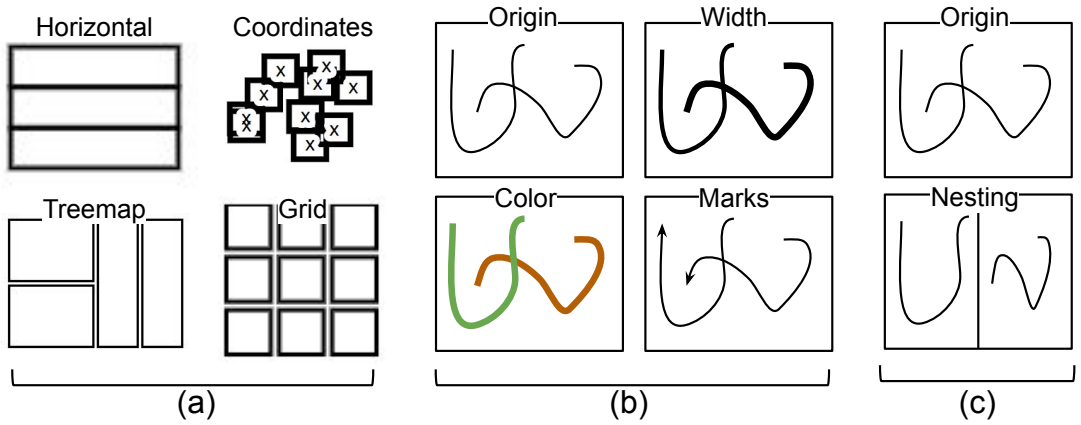

Figure 4. GRIDIFY provides users with a variety of visual abstractions through space division, visual mapping, and nesting. In space division, the space is divided into cells with different patterns, such as horizontal, coordinates, treemap, and grid, as shown in (a). Visual mapping offers a range of visual formats based on the properties of the flow data, including width, color, and marks, as illustrated in (b). With nesting, users can create new hierarchies to represent the flow data in cells generated from the parents' placeholder positions, as demonstrated in (c).

radial, brick

- **Visual mapping:** is the customization of cells using an attribute mapped to its properties, based on the grid type, as shown in Figure 4 (b).
  Parameters: coordinates, height, width, order, color, filling with marks such as circles and lines
- **Nesting:** all grids can be nested with each other and children inherit from the parents' placeholders positions and dimensions, as shown in Figure 4 (c).
  Ex: ⋈ (scatterplot of matrices)

We advocate for the decoupling of abstractions to be more widely supported by interactive techniques. Currently, there is a lack of tools that allow for operating on both directions of the Infovis pipeline at an abstract level. Specifically, we need tools that can provide grid patterns and advanced nesting capabilities for deep hierarchical layouts, which are fundamental operations at both the data and visual levels. While graphical notations like (Slingsby et al., 2009; Satyanarayan et al., 2017; Vuillemot & Boy, 2018).

4. GRIDIFY Tool

Our GRIDIFY framework implements both the data and graphical abstractions introduced in the previous section. It is available as an online web application designed to help analysts build abstract, multidimensional gridded generalization of OD maps. It presents all the parameters for building the necessary data and visual abstractions on a compact *query panel* (Figure 5), which analysts can use to rapidly explore implicit properties in the data, while preserving references to their explicit properties in a *main view* (Figure 5). This section outlines the design principles behind both the query panel and main view, as well as their implementation. We also detail the development of intuitive cues that allow for navigation between pre-set and historical configurations. To provide a comprehensive understanding of the system, we present a typical workflow using GRIDIFY.

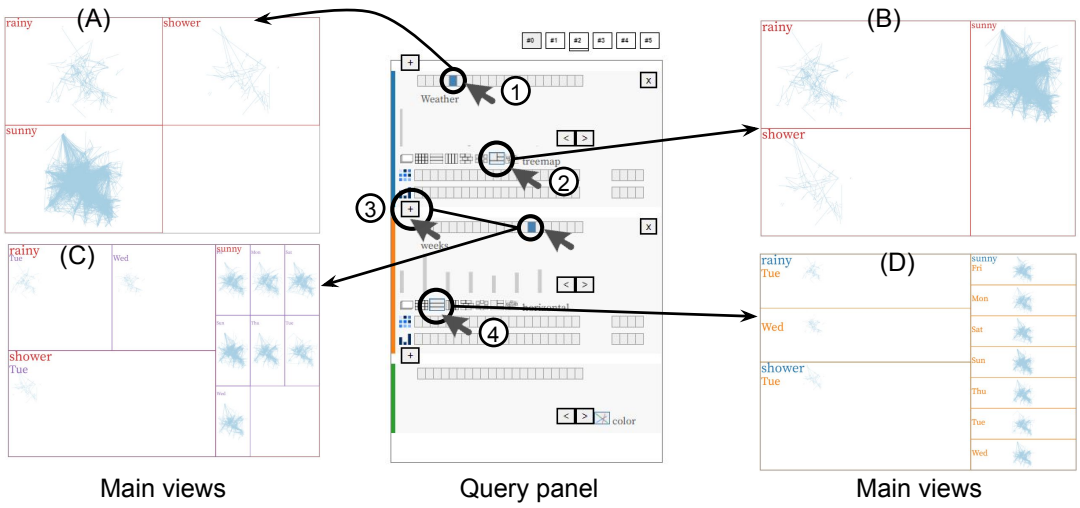

Figure 5. An overview of GRIDIFY implementations and interactions of construct data and visual abstraction. When users browse through datasets, they can select a specific dimension, such as "Weather," to group entities based on the chosen attribute. This process is illustrated in ① and generates a main view that includes different weather conditions, such as rainy, shower, and sunny, as shown in (A). In addition, users can customize the grid patterns, such as a treemap, by fixing the positions of cells through ②, as demonstrated in (B). Finally, clicking on ③ and ④ creates another dimension based on the nesting method, providing users with a more comprehensive view of the data.

4.1. Typical Workflow

In Figure 5, we present a straightforward workflow for users that begin with an empty query sequence. The query panel allows users to create main view divisions and refine them progressively in just a few steps:

(1) *Browsing* all the **dimensions** of a dataset and *picking one* which defines a **grouping**, as shown in Figure 5 (①). For each element of the **domain** cells are created on the main view, as shown in Figure 5 (A);

(2) *Customizing grouping and grids patterns* (Figure 5 ②) consists in defining the dimension domain, unique values (if categorical dimension) or its bins (if quantitative dimension). The cell in main view will change **position** and **dimensions** based on this gridding pattern, as shown in Figure. 5 (B);

(3) *Nesting* by repeating the sequence on another dimension (Figure 5 (④) and (D)). This enables the iterative construction of complex queries and grids by leveraging **dependencies** that occur since they are chained together.

To facilitate efficient exploration, our interface enables users to quickly hover over elements, with immediate updates in the visual space. These updates are transient and become persistent only upon clicking. This approach employs a scrubbing-like interaction that uses sequential feedforward to allow for instant previews and support the reflective construction of complex queries. We use color and position of the widget to convey the state of the query, enabling users to scan the complete list of widgets used for nesting and read sequentially the transformation that was operated, as well as the resulting query.

4.2. Query Panel Design

The query panel (Figure 5-center Figure 6) is a crucial component of GRIDIFY that enables users to query the implicit properties present in the data. The design of this panel is based on several fundamental EDA tool design principles. One of these principles is the use of visual

representations to enhance data analysis, which has been widely discussed in the literature (Wongsuphasawat et al., 2016; Tufte, 1983; Slingsby et al., 2009). Additionally, the query panel incorporates a grid pattern to aid in data exploration and analysis. This feature builds on the principle of using a grid-based layout, which has been found to be effective in organizing information and reducing clutter in data visualizations. Furthermore, the design of the query panel generates hierarchical grid-based layouts. The panel provides a comprehensive and efficient way to visualize geospatial data and derive insights from it. We summarize the principles as follows:

P1 **Make nesting chainable, and show them in a compact way**: the query panel should facilitate building sequential queries that can be chained and viewed together, as well as allow for their tweaking and reorganization, to help analysts better understand the state of their query.

P2 **Expose the data and visual abstractions:** all data dimensions, grouping and aggregation parameters as well as the visual abstraction parameters should be readily visible in the query panel, and not hidden in menus or further folded panels, immediately exposing the breadth of options offered to the analyst.

P3 **Enable constructive selection and constant preview**: the query panel should enable a tight action–feedback loop, and the main mode of interaction should favor continuous actions (e.g. hovering or brushing) over sequential ones (e.g. dropdown selection) providing instant feedback.

P4 **Ensure generic and independent manipulation of the data and visual abstractions**: analysts should be able to query implicit properties of the data independently from their visual abstraction, and reciprocally, they should be able to build visual mappings even if the necessary data abstractions are not defined.

The query panel is composed of rows controlling each level of nesting (shown in Figure 6), both in the data and visual abstractions (**P1**), which display a list of all implicit properties (shown as small rectangular boxes ①), and of grid patterns that can be used to visually encode them (shown in ②). This ensures all data and visual attributes, operations, and parameters are immediately exposed (**P2**). Selecting a grid pattern, and simply brushing across the list of implicit properties will automatically update the main view (**P3**), breaking down the base OD node-link diagram (i. e. the visualization of explicit properties) into a multitude of cells, related to the selected implicit property.

To enable *Grouping* methods in the data abstraction, we display a univariate distribution of each quantitative implicit propertys' value (in the form of a histogram shown as ③), as well as an option for changing bins values. Note however that the *Aggregation* operations can only be applied to quantitative implicit properties (shown in ④). This encoding is relatively compact (**P1**), so many nested operations can be listed.

There is no pre-defined flow for query construction, the user can start with the visual abstraction and then continue with the data abstraction (**P4**) or vice-versa. The operations done at a level of nesting are chained with those of the levels that precede it (**P1**). As a result, the query panel plays an important role in the EDA process (beyond exposing data abstractions): it indicates all current implicit property selections and their visual mappings, and these can be dynamically updated, like in an (interactive) legend (Riche et al., 2010). This means there is no need for additional navigation elements in the interface.

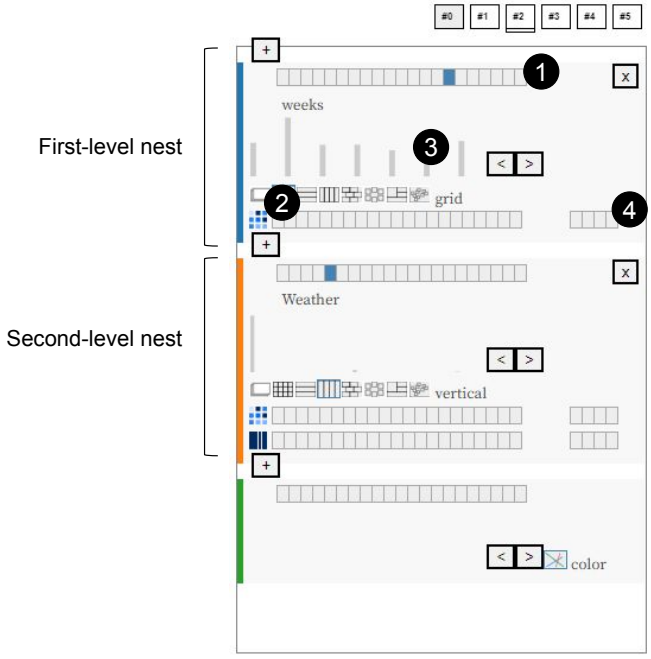

Figure 6. Query panel overview. ① shows the rectangular boxes view all the implicit relations. Grid pattern selection generates different space divisions (e. g., grid, horizontal, or vertical) in ②. The histogram displays the univariate distribution of each quantitative implicit relations' value in ③, and ④ displays options of quantitative implicit relations calculation.

4.3. Main View Design

The main view consists of a canvas on which the explicit properties and all their consecutive subdivisions are rendered according to the constructive design of the query panel. Its design follows one main principle, summarized as follows:

P5 Maintain consistent encoding: A consistent encoding should be preserved (e. g., using position), to allow a focus on attributes' structure, and generate interesting sub-sets. Also browsing should be informative on the attributes space, domains and provide flexibility especially when binning or clustering is needed.

The canvas initially displays the explicit properties. Once a query is added using the query panel, the canvas turns into a hierarchical structure, in which the visualization of explicit properties is broken down into cells, each containing a subset of the explicit properties, relevant to the selected implicit property. Additional encodings can be set, such as a bivariate color scale (Y. Liu & Heer, 2018) to convey **aggregation** values (e. g., `MEAN`, or `COUNT`). For example, if color and grid are selected, a small glyph can be displayed, or a heatmap depending on the level being colored, and the space available. Color encodings of the whole cell can also be used as the level of nesting increases, for performance reasons to reduce the number of elements to draw.

Analysts can then select cells for grouping purposes, by clicking and dragging in the main view. Selecting cells updates a `SELECTED` attribute in the dataset. This attribute can be used to 1) show only the selected values, 2) hide them, or 3) grouping purposes in the query pipeline, e.g. computing aggregate values on the selection, or adding a level of nesting.

5. Implementation Details

We implemented GRIDIFY using Observable Notebook, a reactive, web-based framework compliant with ES6 JavaScript modules. We used the D3 (Bostock et al., 2011) version 5.9.2 library for data manipulation, and in particular the `d3.group` function for calculating nesting. We also used the `d3-gridding` toolkit (Vuillemot & Boy, 2018)–based on D3–for grid partitions. An interactive prototype is available as supplemental material: `https://observablehq.com/d/10a0f8527c21dcd3`.

We used Python Notebooks to pre-process datasets, which are stored as static CSV files. During our development and tests, we found this stack workable up to 100k elements, using a progressive loading of the marks. Most of the data dimensions were available and derived before loading; we used dynamic aggregation and partition such as C-Kmeans for univariate separation into categories.

Serializing such internal visualization mechanisms, independent from the implementation as Domain-Specific-Languages (DSL) (Heer & Bostock, 2010) is in the line of recent efforts in the infovis community such as Hive (Slingsby et al., 2009), Vega-Lite (Satyanarayan et al., 2017), Ellipsis (Satyanarayan & Heer, 2014) or ATOM (Park et al., 2017), among others.

6. Case Studies

We explored a total of 14 datasets with GRIDIFY, all of which are accessible in the prototype provided in the supplemental material. In this section, we present on 4 datasets to showcase the expressiveness of GRIDIFY, to identify interesting exploration patterns, and to recognize limitations in our design and implementation. We have selected these particular case studies because of the diversity of their explicit and implicit connections, and because authors and their collaborators have expertise in the respective domains knowledge, and have prior experience in creating visualizations with the data. As such, these examples provide valuable insights into the practical applications and challenges of our approach.

6.1. Trade Data

We utilized GRIDIFY to analyze trade flows of 512 products (e. g., fuel, cars) traded between 250 countries from 1965 to 2015. Such dataset shows heterogeneous data types, as well as some derived and inconsistent values (e. g., some values are missing even though this dataset has already been curated by the U.N. and economists (Hausmann & Hidalgo, 2014)). Our process was to answer similar questions available from (Hausmann & Hidalgo, 2014) following the *What, where, and when countries export?* scheme. Figure 7 (①) displays the overview of the dataset (A_s ❀ $\emptyset$), which is highly cluttered. Separation by countries using grid on Figure 7 (②) (A_s ❀ BIN) $\bowtie$ (A_s ▦) preserves geo-connections by countries, but clutter is persistent: the reason is that countries have multiple connections over time (50 years), for each product (512), so potentially more than 2000 lines overlapping.

Where does country A export? Answering this question requires selecting a particular country (e. g., France) by clicking on it, for instance starting from the previous visualization Figure 7. Clicks set the `Show only selected` so one can filter by value (e. g., countries without selection). Then the grid view focuses on the selected country Figure 7 (③), which is again cluttered as we break down an aggregation. So one starts over the abstraction process, similar to the first step, but using another strategy by using geo-grid coordinates ($A_{geogrid}$ ❀ $\emptyset$) to encode partner countries as cells which size reflects exports volume Figigure 7 (④). Following steps explore trade flows over time for export Figure 8 (①) and for import Figure 8 (②). The use of

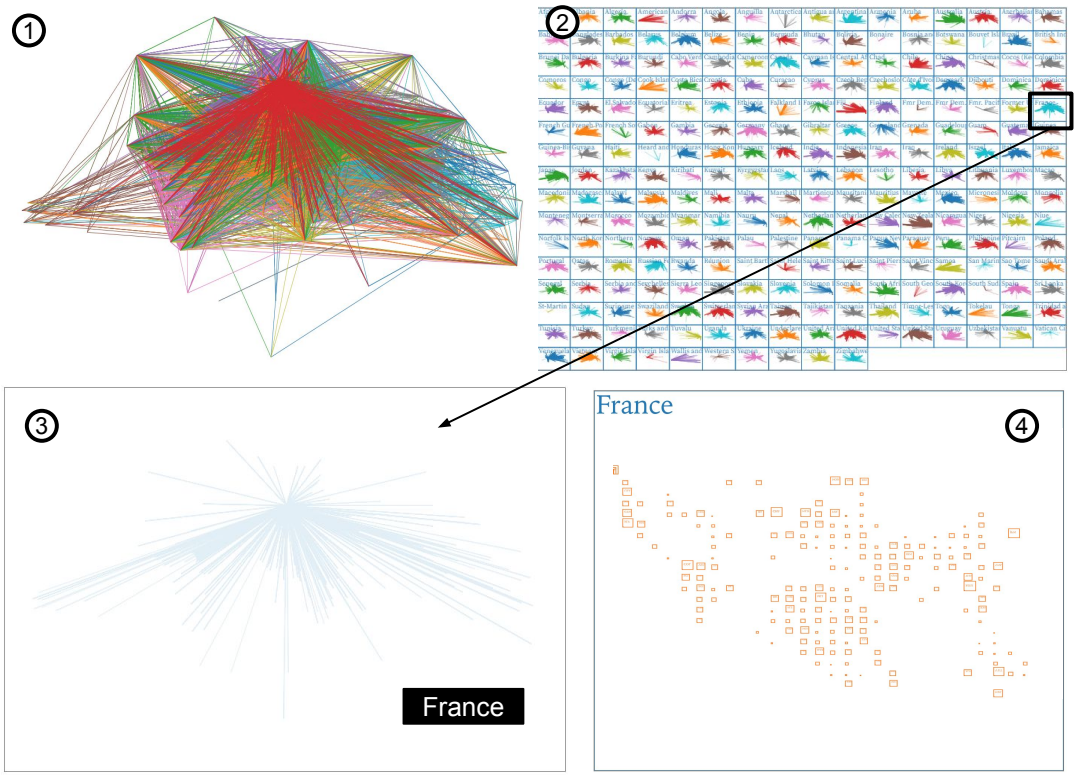

Figure 7. Visualization of GRIDIFY for trade datasets to avoid clutter. An overview of trade datasets all over the world is highly cluttered (①). GRIDIFY allows for the separation of the data by country, which helps to reduce clutter while still providing valuable insights (②). Additionally, users can easily filter data by value by clicking on a specific country, such as France (③). Eventually, one can start over to use geo-grid coordinates to encode partner countries (④).

would be better suited to convey temporal change. By comparing Figure 8 (①) and Figure 8 (②), we can conclude that the chemicals trade of France reached its peak in both export and import. In terms of machinery and vehicles, the export peaked in both 1992 and 1995, while the import only peaked in 1995. Furthermore, France's need for material manufacture was apparent as its imports in that category peaked in 2015. The same exploration process could apply to the series of questions *Which country exports products C?*, *Where does country A export products C? Which products are top exports?*

Overall, we managed to address similar exploration questions than from (Hausmann & Hidalgo, 2014), but in a unified and continuous way. On (Hausmann & Hidalgo, 2014), visualizations are isolated in different pages, and navigation of those visualizations is done using widgets. They have the advantage of quickly aiming at a specific query. Using GRIDIFY the process mostly focused on data manipulations, followed by visual customization, and was done in a constructive way by refining or relaxing questions. Further investigation could be performed on attributes such as landlocked countries, or derivative values such as countries' distance.

6.2. Taxi Data

Large-scale taxi GPS datasets are now publicly available such as the one collected by the city of Wuhan, China, which a co-author of the paper already pre-processed. It contains 7271 entities over a month (Sep. 2013, 145,789 trips). Trajectories for each taxi are available as recorded at a frequency of $1-4$ times per minute, with unique ids and a `STATUS` occupied/vacant/not

Figure 8. The temporal patterns of products reveal the trading activity between France and other countries, both in terms of exports (①) and imports (②).

working/invalid along with fare value when occupied. We derived standard trajectories attributes (Andrienko et al., 2008; Yao et al., 2019; Hochmair, 2016). We performed standard separation, but the dataset is different from the trade: trajectories overlap less as they can more freely in space, while still constrained by roads.

Figure 9 provides an overview of the taxi trajectories in the dataset. The airport, which is a region of interest, is shown in Figure 9 (①). The trajectories originating from the airport tend to be longer than those from other parts of the city, which usually have shorter trips. The hourly distribution of the taxi trajectories, shown in Figure 9 (②), indicates that there are fewer passengers between 2:00 AM and 5:00 AM. One interesting attribute in the dataset flagged the trajectories that corresponded to ”going back to pick up point” during a weekday. Figure 9 (③) displays these trajectories by hour, revealing that most going back trajectories are short, and there are fewer of them overall. To gain a better understanding of the dataset, subsets were created based on the regions where trajectories start with (`cell_group_origin`). Figure 9 (④)

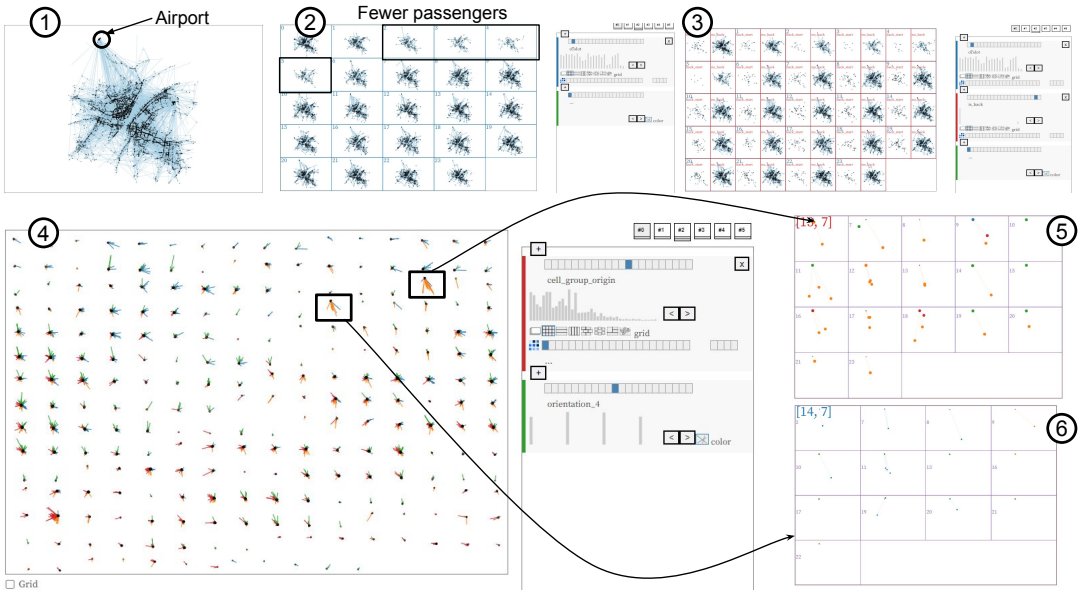

Figure 9. The illustration of taxi trajectories with GRIDIFY provides an overview of the insights gained from analyzing taxi trajectory datasets using the GRIDIFY. It showcases the spatial patterns and points of interest associated with taxi passenger pickups (①), as well as insights into periods of low passenger volumes (②) and whether taxis return to their origin points after drop-offs (③). It also demonstrates the GRIDIFY's ability to divide trajectories into regions based on the cell_group_origins attribute and color-code them based on the orientation_4 attribute, highlighting interesting north-to-south travel patterns (④). The detailed information presented in ⑤ and ⑥ provides a more comprehensive view of the data.

shows these regions, with filled colours indicating the `orientation_4` attribute. The regions highlighted in this figure reveal that almost all the trajectories are from north to south, and these regions are related to the airport. To further explore these regions, the details of the trajectories in the two regions ([13,7] and [14,7]) are shown in Figure 9 (⑤) and Figure 9 (⑥). These figures display the trajectories by hour in each region, revealing details of the temporal patterns of trajectories during a day. By checking the real locations of these two cells, we can confirm that they are related to the airport.

Compared to the general geospatial data visualization shown as Figure 9 (①), the exploration of GRIDIFY enables to better grasp the datasets dimensions distribution by nesting grids based on attributes (Figure 9 (③)). In addition, GRIDIFY enables users to observe details of selected cells, which is shown in Figures 9 (⑤) and (⑥).

6.3. Football Data

The world of sports generates an ever-increasing amount of spatio-temporal connections that can be studied using various tools and techniques. One such tool is footballStories (Perin et al., 2013), which enables the visualization of sequences in football games, such as consecutive passes made by a team, as a clustered graph on the football field. SoccerStories also implements a multi-faceted visualization approach, which divides entities and their connections using a *phase* attribute. In our study, we utilized the same dataset as Opta, which provides data on passes in football games. The dataset contains 12 types of passes, each with up to 50 non-structured *qualifiers* that are specific to each type of pass. We pre-processed the dataset to focus on the most common passes and analyzed a La Liga game between FC Barcelona and Real Madrid (2-2) played on October 7th, 2012. This game comprised a total of 1,622 events.

The coordinate plot of positions Figure 10 (①) first shows all the events but with some data quality issues: entities have missing positions (e. g., no destination entity for a pass, so

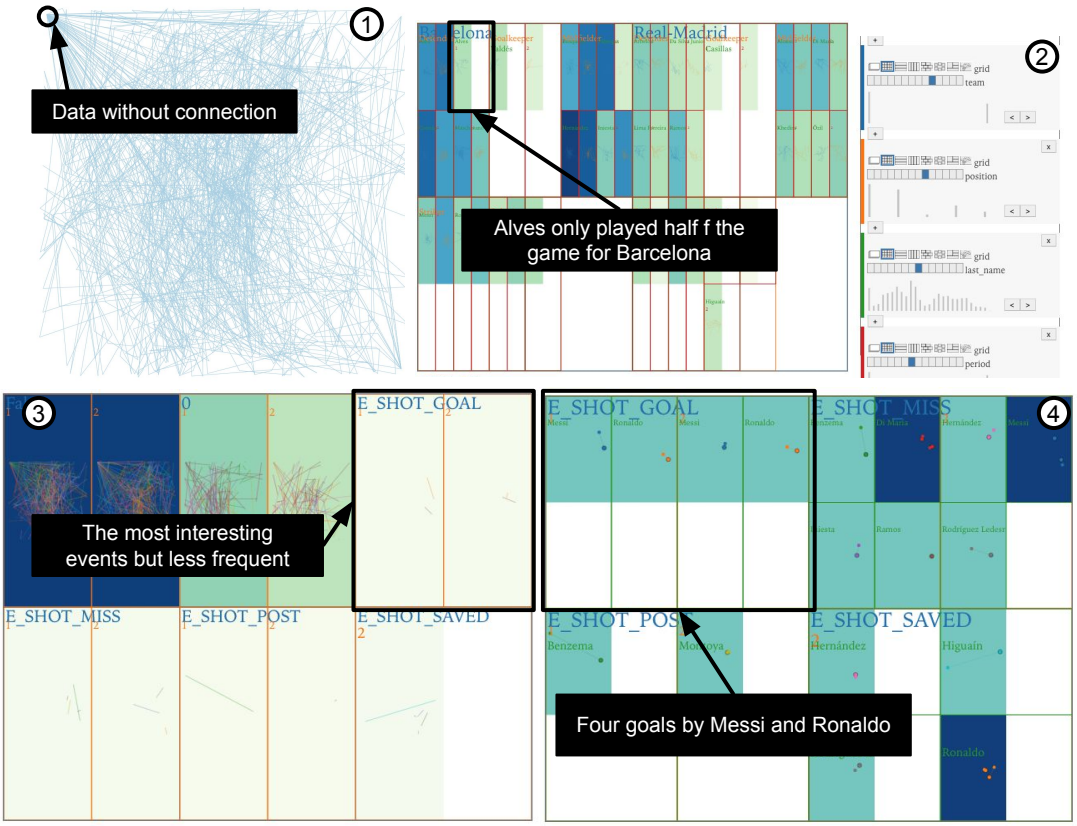

Figure 10. Visualization of football game passes using GRIDIFY. It illustrates the passes made during a football game between two teams consisting of a total of 22 players. It provides insights into the teams' strategies, individual players' pass diversity, and the importance of those passes. The illustration is composed of several components, including an overview of the football flow (①), the number of passes made by players over two time periods (②), highlighting infrequent but significant events (③), and summarizing the most important filtered events (④).

it is set to $(0,0)$). However, as we are interested in connections, we will keep all of them and build hierarchical aggregations, e. g., by the number of events occurrence grouped by `team` $\bowtie$ `role` $\bowtie$ `player` $\bowtie$ `time period` Figure 10 (②). This very simple construction provides a wealth of information not only specific to the game (e. g., Alves only played the first half of the game for Barcelona), but also the style of the teams (Barcelona ball possession is key and they generate many passes), and individual players (midfielders in Barcelona are heavily involved, contrary to Real Madrid). Figure 10 (③) shows however a limit of using GRIDIFY: the most interesting events are not the frequent ones (e. g., goal shots). There only are a few of them, compared to the other types of passes. This is how SoccerStories built sequences: starting from those interesting events which usually are a handful (a dozen by games). But by filtering out to those interesting events, Figure 10 (④) provides a game summary showing the 4 goals of the game (Ronaldo and Messi), and key players who attempted to shoot.

Another limit of GRIDIFY lies in the comparison of cells: to compare shots one may want to orient them in the same direction (e. g., towards the goal from left to right), same for corner passes. Also, there is a lack of metadata such as the football field dimensions, landmarks. Those are key elements implemented in SoccerStories we do not support.

6.4. Public transport accessibility

Transit data is heavily used today for trip planning, but also by urbanists and decision-makers to understand how well transit networks serve the population. We collected a dataset of $45,520$ trips in a major European city at every hour of a given day, an overview as shown in Figure 11 (①). The trips start from 3 distinct locations (origins) and destinations are all reachable areas surrounding the origin for a given time (e. g., 5min) by mode of transport (walk or public transport). We collected various dimensions such as CO_2 emissions. The data comes from the Navitia API: `https://www.navitia.io/`.

The first step was to plot the data in a familiar way when analyzing locations accessibility: as an *isochrone-like maps* as shown in Figure 11 (②). We built it by changing the radius of the destination mark (circle) we emphasized them, which is close to how isochrones are built. We then abstracted space to understand the temporal patterns grouped by `locations (origin_name)` ⋈ `hour`, as shown in Figure 11 (③). Figure 11 (③) groups each of the locations as a row and a vertical grid divides by hour. The central part of the grids shows a (small) spatial information of transit accessibility for each hour. The background colors the SUM of trips so we can quickly spot the time of the day the public transport network is most active (during morning and afternoon rush hours), but also the lack of rapid transit mechanism during the night. We can refine the query by removing the walked journeys (another division) and dividing again by distance grouped by `distance` ⋈ `locations(origin_name)` ⋈ `hour`, this leads to locations having the best and most consistent large reach (something not conveyed by isochrone maps), as shown in Figure 11 (④). By modifying two steps of the pipeline grouped by `duration` ⋈ `locations(origin_name)` ⋈ `direction(outgoing and incoming journeys)` ⋈ `hour` from Figure 11 (④), we compare the duration of outgoing and incoming journeys (another division), and identify which location has the most asymmetry (adding division on the time difference at the root using `BINNING` of *Grouping* method, as mentioned in Section 3.1), as shown in Figure 11 (⑤). Finally, starting from scratch again, partitioning the trajectories by the length of journeys (distance) and origins (grouped by `locations(origin_name)` ⋈ `distance`), and coloring the marks by journey duration, reveals the directions that are easier to reach than others, and highlights the difference between distance and travel time, as shown in Figure 11 (⑥).

7. Discussion

Our case studies indicate that the current implementation of GRIDIFY enables complex EDA using a simple set of data and visual abstractions. We discuss here the implications of our abstractions, their implementation in GRIDIFY, and the challenges it raises.

7.1. Expressiveness and Applicability

GRIDIFY is already a very expressive tool despite encoding simple grouping and aggregations mechanisms, and a consistent visual encoding (using circle/marks or color for leaves, and position otherwise as we follow **P5**).

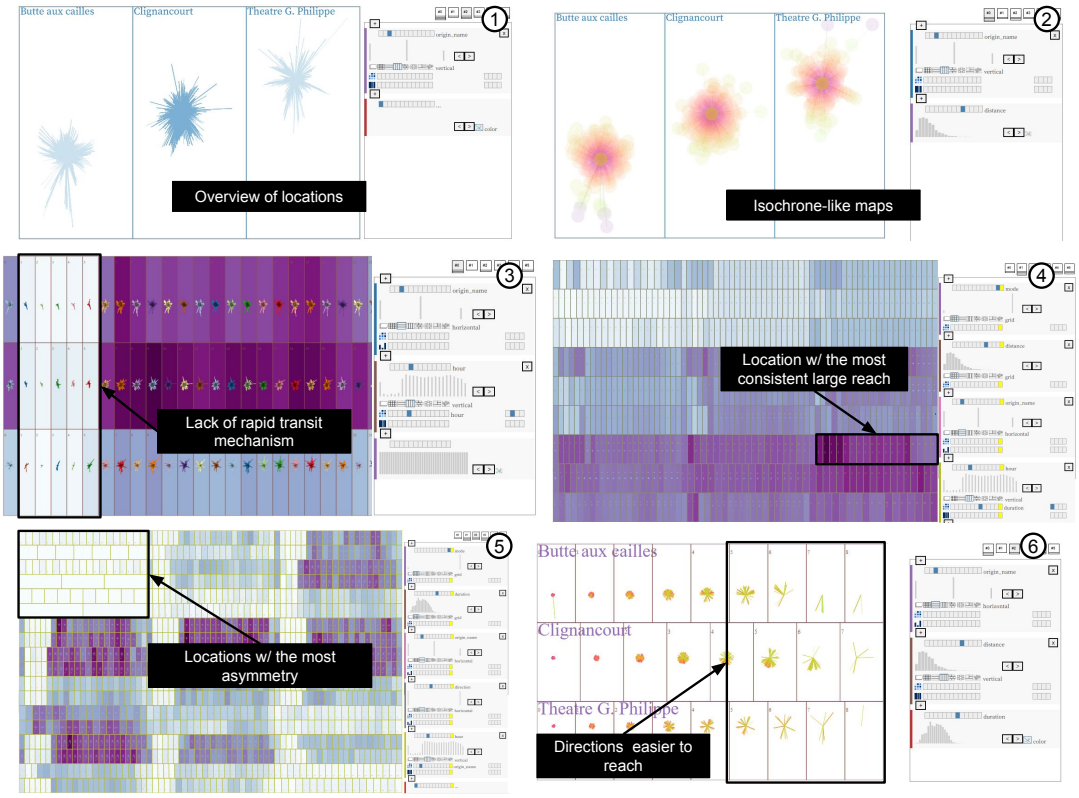

Figure 11. Illustration of transit data visualization in a major European city using GRIDIFY. This visualization provides a clear understanding of the basic accessibility of the transit. It showcases three different locations in ②, which are represented using an isochrone-like map created by adjusting the radius of the destination marks (circles). ③ showcases the temporal patterns by grouping the locations based on hours. The best and most consistent large reach is highlighted in ④. ⑤ illustrates the locations with the most asymmetry, revealing fewer incoming journeys across all three locations. Lastly, ⑥ highlights the directions that are easier to reach, as they provide access to longer distances in the same duration compared to others.

GRIDIFY could handle more data and visual abstractions, such as partitioning, aggregation, and grid patterns. Data aggregation can be integrated on the fly during analysis in the Observable Notebook, by writing functions in an advanced mode. Grid partitions rely on an open-source toolkit–`d3-gridding`–that offers a modular approach to easily add new **grid patterns**. However, in our current approach grids should have a recursive construction mechanism. Hexagonal grids could be an improvement (Yao et al., 2019) as they sometimes provide a better binning estimation, but GRIDIFY only supports rectangular cells. Finally, GRIDIFY is compliant to include Vega specifications at the leaves nodes to render aggregation charts (instead of the marks or the colors). Design implications on the query panel are related to adding chart templates either by the queries or as pre-set configurations.

After analyzing the visual mapping selections made by users in our case studies, we have identified several principles. When encoding temporal dimensions, users tend to prefer a horizontal pattern (▤), whereas for other dimensions, they tend to opt for a vertical pattern (▥), as illustrated in Figure 8. Conversely, when displaying quantitative and categorical dimensions, users often select a grid pattern (▦), as demonstrated in Figure 7 (②). Additionally, the coordinate pattern is well-suited for visualizing geospatial information in each cell, as depicted in Figure 7 (④).

So far we only encoded leaves using either color for aggregation or circles and rectangles for explicit connections. Lines that have a rich design space (Perin et al., 2018) and curve design (Bach et al., 2018) could be used, e. g., to encode local connections properties such

as speed for finer-grain representations. This would extend the **visual mapping** section of the query panel and would probably require a specific legend. Those could be useful at an occupation stage, but for exploratory tasks, one would recommend adding better dynamic opacity techniques (Matejka et al., 2015).

GRIDIFY can be applied to any type of geospatial data, beyond simple connections. For instance, geo-trajectories can be re-constructed grouped by `TRAJECTORY_ID` and ordered over time using the current version of GRIDIFY (see Taxi case studies). New aggregations will be needed to dynamically calculate distances, and other trajectories properties ((Chang et al., 2008)). Figure 9 already shows some cases where a grouping of trajectories can benefit from GRIDIFY. However, a limit of this approach is the number of the segment to display a full resolution trajectory. Also, some metrics on trajectories are relative to the sequence of segments (e. g., sliding window speed) and cannot be derived using aggregation. In a similar fashion, single geo-data points–or even abstract points such as scatterplots– can be used in GRIDIFY without any change to the current version of GRIDIFY.

7.2. Scalability

GRIDIFY's scalability in the number of items is limited by the number of cells and marks it can draw simultaneously. Our prototype managed to handle up to 100k data points (marks) and thousands of rectangles/placeholders. This limit is set by the Observable reactive framework we picked, as it facilitates prototyping and re-use of visualization libraries. Switching to GPU rendering is a classical step that would help but would be limited at some point.

We argue most promising approaches to tackling scalability issues are related to strategies (e. g., domain aggregations, marks aggregation, etc.) to first display aggregation of cells for immediate feedback, and then progressively render details such as nested cells and marks: only the statistical properties of dimensions need to be known in advance (e. g., distribution)—the data points can be loaded later. Regarding the scalability in the number of dimensions, a first limit is on the display of implicit property rectangles and nesting chaining: it was tested with up to 20 implicit properties, but beyond more compact design should be used. For datasets with thousands of implicit properties, adaptive exploration strategies should be developed to suggest/re-order these relations according to the current view (similarly to what is done in EvoGraphDice (Cancino et al., 2012). Dimensions reduction techniques could also be used as an **aggregation** method.

7.3. Perspectives

Our future work is focused on incorporating the latest exploratory analysis features that are available in the InfoVis community. To achieve this goal, we have identified several key areas of improvement as follows:

Multiple views. Exploratory analysis often requires more than a single view, in particular, to provide context. Such context is useful to provide a dataset overview constantly available, e. g., all countries, while a specific country is selected (instead of filtering out all other countries). At the moment, the exploration strategies match the *small multiple, large single* (Elzen & Wijk, 2013) approach where each view is followed by small multiples that provide navigation options.

As GRIDIFY technically and conceptually relies on (Vuillemot & Boy, 2018), multiple static views can be defined in a data-driven manner. So adding a 2-view, Focus+Context layout would require two steps: first at some point to inject a data array with those views properties (which are cells); the linking (shared selection between views) would be provided by the

SELECTED attribute. And then branching the abstractions based on those views (e. g., to assign particular abstractions to the static cells). The impact on the query view would be important as it will not be linear anymore. Techniques like ElasticHierarchies (McGuffin & Chignell, 2005), VisBricks (Lex et al., 2011), TPFlow (D. Liu et al., 2019) and Baobabview (Elzen & Wijk, 2011) are design candidates to improve the sequence with branching. Recently the use of Virtual Reality (Yang et al., 2018) has been proposed to explore Origin-Destinations—but with explicit properties encoded as lines—offering brand new spaces to further explore using a grid-based approach.

Animated transitions. Entities (and their connections) have a continuous representation in GRIDIFY. They can be animated when cells change positions. Similarly, the transition between grid patterns (Wood et al., 2011) provides benefits to users, even though sometimes cells change shape. However, in most cases many parameters change at once: attribute, grid pattern, and aggregation method. Cells may then go through multiple states: appear, disappear, update (i. e. change position and shape). Communicating the change of the cell is an open challenge as similar grids or cells (position and size) do not necessarily encode similar data. A simple approach could be a tree-based animated transitions that collapses/expands nodes based on a hierarchicy (Guilmaine et al., 2012).

We argue GRIDIFY provides an infrastructure to address animation design challenges (related to speed, duration, and timing (Heer & Robertson, 2007)) as it captures exact states of visualizations and can calculate the path towards another one. Designing a better dynamic programming algorithm (than the Levenshtein distance) that optimizes those parameters is promising. However, as animations remains a craft, quantifying its quality needs to automatically assess the number of crossing, and visual complexity of a scene, which may be subjective and dependent on the exploration task (e. g., that may require speed at the expense of quality). The PRE-SET view may reflect the magnitude of such change beyond the number of piling states as such as small multiples (Bach et al., 2015) that encode temporal change using matrices.

Explorations Recommendations. As the exploration space may get very large to explore, recent works suggest ranking visualizations based on variable selections and statistical distributions (Wongsuphasawat et al., 2017, 2016).

Sophisticated encoding and intricate interactions are essential for addressing certain aspects, and therefore, formal design and studies will be required. Additionally, there are open-ended questions that require careful consideration, such as explaining the real position of a position and the meaning of absolute coordinates. These inquiries require a more exploratory approach and may necessitate further investigation.

8. Conclusion

In this article, we presented GRIDIFY, a generalization of OD maps and its implementation as an exploratory data analysis (EDA) tool for primarily geospatial data, which extends previous work on grid-system approaches to visualization design (Vuillemot & Boy, 2018). We have demonstrated its expressiveness and effectiveness through four case studies. The tool builds on a core concept of decoupling data transformations and visualization construction mechanisms. Its main advantages include maintaining an explicit encoding of both data and visual parameters, which provides analysts with an overview of all their options, which they can rapidly test; chaining nesting operations that are rendered in a grid view that provides constant feedback, allowing analysts to select elements on which they want to zoom in; and exporting and loading pre-set configurations, which provides a mechanism for transitioning between views in a step-by-step way, for pre-defining the exploration charts and patterns, and potentially

for recommending future explorations steps. To finish, while our design and implementation of GRIDIFY has so far focused mainly on OD data and geospatial data, essentially because Authors work a lot with this type of data, we strongly believe the tool can handle a wider variety of data and visual abstractions and transitions. We intend to explore these possibilities in the future.

References

Amar, R., Eagan, J., & Stasko, J. (2005). Low-Level Components of Analytic Activity in Information Visualization. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization* (pp. 15–). Washington, DC, USA: IEEE Computer Society. Retrieved 2014-08-21, from http://dx.doi.org/10.1109/INFOVIS.2005.24

Andrienko, N., Andrienko, G., Pelekis, N., & Spaccapietra, S. (2008). Basic Concepts of Movement Data. In F. Giannotti & D. Pedreschi (Eds.), *Mobility, Data Mining and Privacy: Geographic Knowledge Discovery* (pp. 15–38). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved 2019-03-24, from https://doi.org/10.1007/978-3-540-75177-9$_2$

Bach, B., Henry-Riche, N., Dwyer, T., Madhyastha, T., Fekete, J.-D., & Grabowski, T. (2015). Small MultiPiles: Piling Time to Explore Temporal Patterns in Dynamic Networks. *Computer Graphics Forum*, *34*(3), 31–40. Retrieved 2019-03-31, from https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12615

Bach, B., Perin, C., Ren, Q., & Dragicevic, P. (2018, April). Ways of Visualizing Data on Curves. In (pp. 1–14). Retrieved 2019-03-17, from https://hal.inria.fr/hal-01818137/document

Bau, O., & Mackay, W. E. (2008). OctoPocus: A Dynamic Guide for Learning Gesture-based Command Sets. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (pp. 37–46). New York, NY, USA: ACM. Retrieved 2019-03-27, from http://doi.acm.org/10.1145/1449715.1449724

Beecham, R., Rooney, C., Meier, S., Dykes, J., Slingsby, A., Turkay, C., ... Wong, B. L. W. (2016, June). Faceted Views of Varying Emphasis (FaVVEs): a framework for visualising multi-perspective small multiples. *Computer Graphics Forum*, *35*(3), 241–249. Retrieved 2019-03-25, from https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12900

Bostock, M., Ogievetsky, V., & Heer, J. (2011, December). D3; Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2301–2309.

Boy, J., Eveillard, L., Detienne, F., & Fekete, J.-D. (2016, January). Suggested Interactivity: Seeking Perceived Affordances for Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, *22*(1), 639–648. Retrieved 2019-03-31, from https://hal.inria.fr/hal-01188973/document

Cancino, W., Boukhelifa, N., & Lutton, E. (2012, June). EvoGraphDice: Interactive evolution for visual analytics. In *2012 IEEE Congress on Evolutionary Computation* (pp. 1–8).

Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann.

Chang, H.-w., Tai, Y.-c., Chen, H.-w., & Hsu, J. Y.-j. (2008). iTaxi: Context-Aware Taxi Demand Hotspots Prediction Using Ontology and Data Mining Approaches. , 8.

Cleveland, W. S., & McGill, R. (1984). Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association*, *79*(387), 531–554. Retrieved 2019-03-24, from https://www.jstor.org/stable/2288400

Dodge, S., Weibel, R., Ahearn, S. C., Buchin, M., & Miller, J. A. (2016, May). Analysis of movement data. *International Journal of Geographical Information Science*, *30*(5), 825–834. Retrieved 2019-03-17, from https://doi.org/10.1080/13658816.2015.1132424

Ellis, G., & Dix, A. (2007, November). A Taxonomy of Clutter Reduction for Information Visualisation. *IEEE Transactions on Visualization and Computer Graphics*, *13*(6), 1216–1223.

Elmqvist, N., Dragicevic, P., & Fekete, J. (2008, November). Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. *IEEE Transactions on Visualization and Computer Graphics*, *14*(6), 1539–1148.

Elmqvist, N., & Fekete, J.-D. (2010, May). Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE Transactions on Visualization and Computer Graphics*, *16*(3), 439–454. Retrieved 2019-01-01, from http://ieeexplore.ieee.org/document/5184827/

Elzen, S. v. d., & Wijk, J. J. v. (2011, October). BaobabView: Interactive construction and analysis of decision trees. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)* (pp. 151–160).

Elzen, S. v. d., & Wijk, J. J. v. (2013). Small Multiples, Large Singles: A New Approach for Visual Data Exploration. *Computer Graphics Forum*, *32*(3pt2), 191–200. Retrieved 2019-03-17, from https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12106

Elzen, S. v. d., & Wijk, J. J. v. (2014, December). Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations. *IEEE Transactions on Visualization and Computer Graphics*, *20*(12), 2310–2319.

Ersoy, O., Hurter, C., Paulovich, F., Cantareiro, G., & Telea, A. (2011, December). Skeleton-Based Edge Bundling for Graph Visualization. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2364–2373.

Furmanova, K., Gratzl, S., Stitz, H., Zichner, T., Jaresova, M., Ennemoser, M., … Streit, M. (2017, December). Taggle: Scalable Visualization of Tabular Data through Aggregation. *arXiv:1712.05944 [cs]*. Retrieved 2019-01-28, from http://arxiv.org/abs/1712.05944

Ghoniem, M., Fekete, J., & Castagliola, P. (2004, October). A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. In *IEEE Symposium on Information Visualization* (pp. 17–24).

Guilmaine, D., Viau, C., & McGuffin, M. J. (2012). Hierarchically Animated Transitions in Visualizations of Tree Structures. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (pp. 514–521). New York, NY, USA: ACM. Retrieved 2019-01-28, from http://doi.acm.org/10.1145/2254556.2254653

Hausmann, R., & Hidalgo, C. A. (2014). *The atlas of economic complexity: Mapping paths to prosperity*. MIT Press.

Heer, J., & Bostock, M. (2010). Declarative Language Design for Interactive Visualization. *IEEE Transactions on Visualization and Computer Graphics*, *16*(6), 1149–1156.

Heer, J., & Robertson, G. (2007, November). Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics*, *13*(6), 1240–1247.

Hochmair, H. H. (2016, January). Spatiotemporal Pattern Analysis of Taxi Trips in New York City. *Transportation Research Record*, *2542*(1), 45–56. Retrieved 2019-03-31, from https://doi.org/10.3141/2542-06

Holten, D., & Van Wijk, J. J. (2009). Force-Directed Edge Bundling for Graph Visualization. *Computer Graphics Forum*, *28*(3), 983–990. Retrieved 2022-06-03, from

https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01450.x

Hurter, C., Riche, N. H., Drucker, S. M., Cordeil, M., Alligier, R., & Vuillemot, R. (2018, October). FiberClay: Sculpting Three Dimensional Trajectories to Reveal Structural Insights. *IEEE Transactions on Visualization and Computer Graphics*, *25*(1), 704–714.

Iqbal, M. S., Choudhury, C. F., Wang, P., & González, M. C. (2014, March). Development of origin–destination matrices using mobile phone call data. *Transportation Research Part C: Emerging Technologies*, *40*, 63–74. Retrieved 2019-03-31, from http://www.sciencedirect.com/science/article/pii/S0968090X14000059

Jenny, B., Stephen, D. M., Muehlenhaus, I., Marston, B. E., Sharma, R., Zhang, E., & Jenny, H. (2017). Force-directed layout of origin-destination flow maps. *International Journal of Geographical Information Science*, *31*(8), 1521–1540. Retrieved from http://dx.doi.org/10.1080/13658816.2017.1307378 (Publisher: Taylor & Francis)

Lex, A., Schulz, H., Streit, M., Partl, C., & Schmalstieg, D. (2011, December). VisBricks: Multiform Visualization of Large, Inhomogeneous Data. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2291–2300.

Li, D., Lin, Y., Zhao, X., Song, H., & Zou, N. (2011). Estimating a Transit Passenger Trip Origin-Destination Matrix Using Automatic Fare Collection System. In J. Xu, G. Yu, S. Zhou, & R. Unland (Eds.), *Database Systems for Adanced Applications* (pp. 502–513). Springer Berlin Heidelberg.

Li, Q., Bao, X., Song, C., Zhang, J., & North, C. (2003, April). Dynamic query sliders vs. brushing histograms. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (pp. 834–835). New York, NY, USA: Association for Computing Machinery. Retrieved 2024-01-19, from https://dl.acm.org/doi/10.1145/765891.766020

Liu, D., Xu, P., & Ren, L. (2019, January). TPFlow: Progressive Partition and Multidimensional Pattern Extraction for Large-Scale Spatio-Temporal Data Analysis. *IEEE Transactions on Visualization and Computer Graphics*, *25*(1), 1–11.

Liu, Y., & Heer, J. (2018). Somewhere Over the Rainbow: An Empirical Assessment of Quantitative Colormaps. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18* (pp. 1–12). Montreal QC, Canada: ACM Press. Retrieved 2019-03-27, from http://dl.acm.org/citation.cfm?doid=3173574.3174172

Matejka, J., Anderson, F., & Fitzmaurice, G. (2015). Dynamic Opacity Optimization for Scatter Plots. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 2707–2710). New York, NY, USA: ACM. Retrieved 2019-03-31, from http://doi.acm.org/10.1145/2702123.2702585

McGuffin, a. M. J., & Chignell, M. H. (2005, October). Elastic hierarchies: combining treemaps and node-link diagrams. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* (pp. 57–64).

Mihalisin, T., Gawlinski, E., Timlin, J., & Schwegler, J. (1990). Visualizing a scalar field on an n-dimensional lattice. In *Proceedings of the First IEEE Conference on Visualization: Visualization90* (pp. 255–262). IEEE.

Mihalisin, T., Timlin, J., & Schwegler, J. (1991). Visualizing multivariate functions, data, and distributions. *IEEE Computer Graphics and Applications*, *11*(03), 28–35. (Publisher: IEEE Computer Society)

Moscovich, T., Chevalier, F., Henry, N., Pietriga, E., & Fekete, J.-D. (2009). Topology-aware Navigation in Large Networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2319–2328). New York, NY, USA: ACM. Retrieved 2019-03-31, from http://doi.acm.org/10.1145/1518701.1519056

Munizaga, M. A., & Palma, C. (2012, October). Estimation of a disaggregate multimodal public transport Origin–Destination matrix from passive smartcard data from Santiago, Chile. *Transportation Research Part C: Emerging Technologies*, *24*, 9–18. Retrieved 2019-03-31, from http://www.sciencedirect.com/science/article/pii/S0968090X12000095

Munzner, T. (2014). *Visualization analysis and design*. AK Peters/CRC Press.

Norman, D. A. (2002). *The Design of Everyday Things*. New York, NY, USA: Basic Books, Inc.

Park, D., Drucker, S. M., Fernandez, R., & Elmqvist, N. (2017). ATOM: A Grammar for Unit Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, *PP*(99), 1–1.

Perin, C., Vuillemot, R., & Fekete, J.-D. (2013, December). SoccerStories: A Kick-off for Visual Soccer Analysis. *IEEE Transactions on Visualization and Computer Graphics*, *19*(12), 2506–2515.

Perin, C., Wun, T., Pusch, R., & Carpendale, S. (2018, January). Assessing the Graphical Perception of Time and Speed on 2D+Time Trajectories. *IEEE Transactions on Visualization and Computer Graphics*, *24*(1), 698–708.

Rae, A. (2009). From spatial interaction data to spatial interaction information? Geovisualisation and spatial structures of migration from the 2001 UK census. *Computers, Environment and Urban Systems*, *33*(3), 161–178. Retrieved from http://dx.doi.org/10.1016/j.compenvurbsys.2009.01.007 (Publisher: Elsevier Ltd)

Riche, N. H., Lee, B., & Plaisant, C. (2010). Understanding Interactive Legends: a Comparative Evaluation with Standard Widgets. *Computer Graphics Forum*, *29*(3), 1193–1202.

Roelandt, N., Bahoken, F., Le Campion, G., Jégou, L., Maisonobe, M., Mericskay, B., & Côme, E. (2021, August). ONE ARABESQUE IN THE SMALL WORLD OF OD WEBMAPS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVI-4-W2-2021*, 147–154. Retrieved 2024-01-19, from https://isprs-archives.copernicus.org/articles/XLVI-4-W2-2021/147/2021/ (Conference Name: ISPRS WG IV/4, WG IV/9 & WG V/8
FOSS4G 2021 – Academic Track - 27 September–2 October 2021, Buenos Aires, Argentina Publisher: Copernicus GmbH)

Satyanarayan, A., & Heer, J. (2014, June). Authoring Narrative Visualizations with Ellipsis. *Computer Graphics Forum*, *33*(3), 361–370. Retrieved 2014-11-15, from http://onlinelibrary.wiley.com/doi/10.1111/cgf.12392/abstract

Satyanarayan, A., Moritz, D., Wongsuphasawat, K., & Heer, J. (2017, January). Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics*, *23*(1), 341–350. Retrieved 2017-02-03, from https://doi.org/10.1109/TVCG.2016.2599030

Satyanarayan, A., Wongsuphasawat, K., & Heer, J. (2014). Declarative Interaction Design for Data Visualization. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (pp. 669–678). New York, NY, USA: ACM. Retrieved 2019-03-31, from http://doi.acm.org/10.1145/2642918.2647360

Scheepens, R., Hurter, C., Van De Wetering, H., & Van Wijk, J. J. (2016, January). Visualization, Selection, and Analysis of Traffic Flows. *IEEE Transactions on Visualization and Computer Graphics*, *22*(1), 379–388.

Schon, D. (1984). *The Reflective Practitioner: How Professionals Think In Action*. Basic Books.

Slingsby, A., Dykes, J., & Wood, J. (2009, November). Configuring Hierarchical Layouts

to Address Research Questions. *IEEE Transactions on Visualization and Computer Graphics*, *15*(6), 977–984.

Stolte, C., Tang, D., & Hanrahan, P. (2003, April). Multiscale visualization using data cubes. *IEEE Transactions on Visualization and Computer Graphics*, *9*(2), 176–187.

Stolte, C., Tang, D., & Hanrahan, P. (2008, November). Polaris: A System for Query, Analysis, and Visualization of Multidimensional Databases. *Commun. ACM*, *51*(11), 75–84. Retrieved 2017-02-09, from http://doi.acm.org/10.1145/1400214.1400234

Sun, S. (2019). A spatial one-to-many flow layout algorithm using triangulation, approximate Steiner trees, and path smoothing. *Cartography and Geographic Information Science*, *46*(3), 243–259. Retrieved from https://doi.org/10.1080/15230406.2018.1437359 (Publisher: Taylor & Francis)

Tennekes, M., & Chen, M. (2021). Design Space of Origin-Destination Data Visualization. *Computer Graphics Forum*, *40*(3), 323–334. Retrieved 2024-01-10, from https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14310 (_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14310)

Terry, M., & Mynatt, E. D. (2005, December). Enhancing general-purpose tools with multi-state previewing capabilities. *Knowledge-Based Systems*, *18*(8), 415–425. Retrieved 2019-03-26, from http://www.sciencedirect.com/science/article/pii/S0950705105000730

Tobler, W. R. (1987). Experiments in migration mapping by computer. *The American Cartographer*, 155–163.

Tufte, E. (1983). *The Visual Display of Quantitative Information* (2nd ed.). Graphics Press.

Tukey, J. W. (1977). Exploratory data analysis. *Reading, Ma*, *231*, 32.

Vermeulen, J., Luyten, K., van den Hoven, E., & Coninx, K. (2013). Crossing the Bridge over Norman's Gulf of Execution: Revealing Feedforward's True Identity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1931–1940). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2470654.2466255

Vuillemot, R., & Boy, J. (2018, January). Structuring Visualization Mock-Ups at the Graphical Level by Dividing the Display Space. *IEEE Transactions on Visualization and Computer Graphics*, *24*(1), 424–434. Retrieved 2018-12-02, from http://ieeexplore.ieee.org/document/8017650/

Wattenberg, M. (2006). Visual Exploration of Multivariate Graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 811–819). New York, NY, USA: ACM. Retrieved 2019-03-31, from http://doi.acm.org/10.1145/1124772.1124891

Wilkinson, L. (2005). *The Grammar of Graphics (Statistics and Computing)*. Berlin, Heidelberg: Springer-Verlag.

Willett, W., Heer, J., & Agrawala, M. (2007). Scented widgets: Improving navigation cues with embedded visualizations. In *In Proc. of of the SIGCHI Conference on Human Factors in Computing Systems. 9* (pp. 51–58).

Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., & Heer, J. (2016, January). Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics*, *22*(1), 649–658.

Wongsuphasawat, K., Qu, Z., Moritz, D., Chang, R., Ouk, F., Anand, A., ... Heer, J. (2017). Voyager 2: Augmenting Visual Analysis with Partial View Specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Com-*

puting Systems (pp. 2648–2659). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/3025453.3025768

Wood, J., Dykes, J., & Slingsby, A. (2010, May). Visualisation of Origins, Destinations and Flows with OD Maps. *CARTOGR J*, *47*, 117–129. Retrieved 2018-11-05, from http://discovery.ucl.ac.uk/1330588/

Wood, J., Slingsby, A., & Dykes, J. (2011, December). Visualizing the Dynamics of London's Bicycle-Hire Scheme. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *46*(4), 239–251. Retrieved 2022-05-12, from https://utpjournals.press/doi/10.3138/carto.46.4.239

Yang, Y., Dwyer, T., Jenny, B., Marriott, K., Cordeil, M., & Chen, H. (2018). Origin-destination flow maps in immersive environments. *IEEE transactions on visualization and computer graphics*, *25*(1), 693–703. (Publisher: IEEE)

Yao, X., Wu, L., Zhu, D., Gao, Y., & Liu, Y. (2019, January). Visualizing spatial interaction characteristics with direction-based pattern maps. *Journal of Visualization*. Retrieved 2019-03-17, from https://doi.org/10.1007/s12650-018-00543-4

Zhao, J., Collins, C., Chevalier, F., & Balakrishnan, R. (2013, December). Interactive Exploration of Implicit and Explicit Relations in Faceted Datasets. *IEEE Transactions on Visualization and Computer Graphics*, *19*(12), 2080–2089.

Zhu, X., Guo, D., Koylu, C., & Chen, C. (2019, September). Density-based multi-scale flow mapping and generalization. *Computers, Environment and Urban Systems*, *77*, 101359. Retrieved 2022-06-03, from https://www.sciencedirect.com/science/article/pii/S0198971518305714