DOI: 10.1111/cgf.14310
Eurographics Conference on Visualization (EuroVis) 2021
R. Borgo, G. E. Marai, and T. von Landesberger
(Guest Editors)

Volume 40 (2021), Number 3

Design Space of Origin-Destination Data Visualization

Martijn Tennekes[1] and Min Chen[2]

[1]Statistics Netherlands, the Netherlands and [2]University of Oxford, United Kingdom

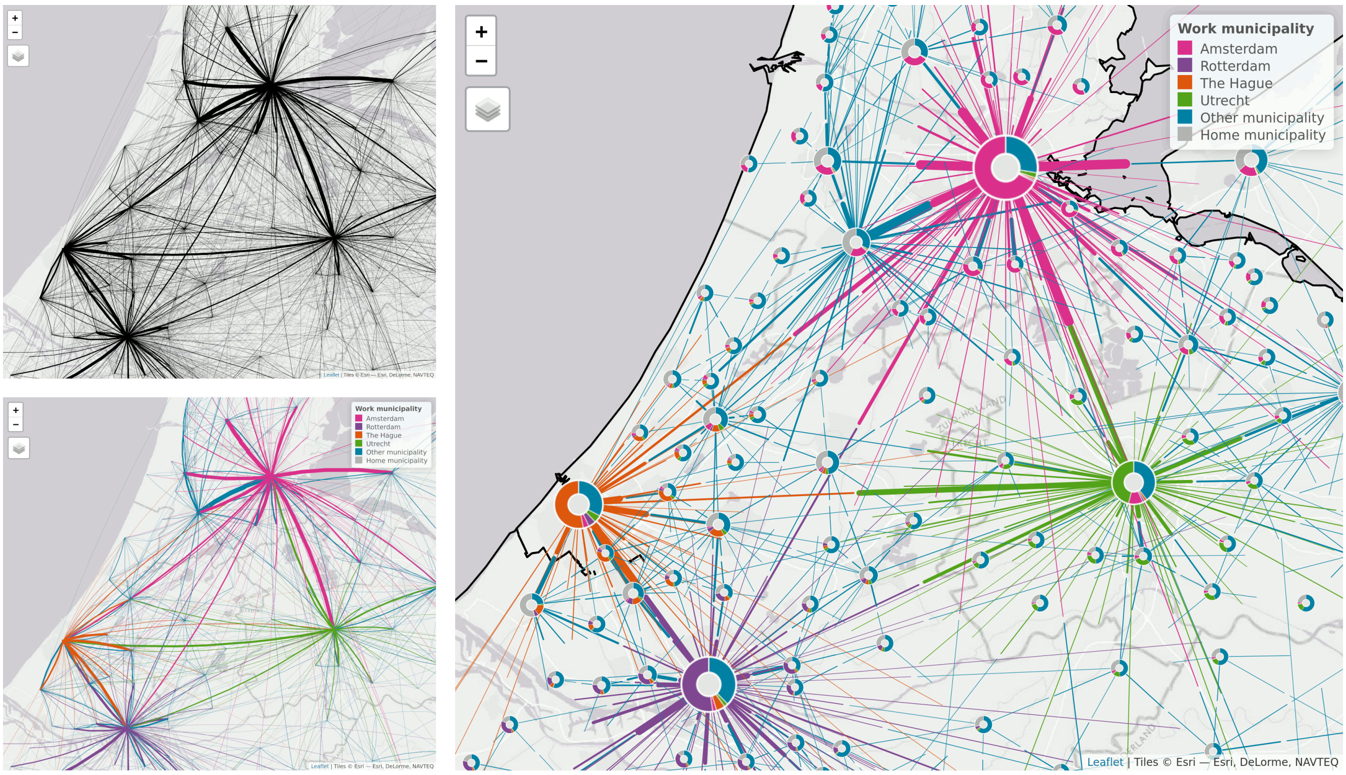

Figure 1: *Visualizing the commuting traffic flows in the Netherlands.* Top-left: *A naive visual encoding attempts to convey every piece of information in the dataset but a viewer can only gain a rough impression about the traffic density.* Bottom-left: *Filtering and color-encoding offers some improvement.* Right: *Our ODDV design space helps us find a design where each edge is drawn from the half-way point towards the destination, so focussing on the incoming traffic, while the outgoing traffic is summarized with a small doughnut-chart at each origin. Although it "draws" less information than those on the left, more information can be perceived.*

Abstract

Visualization is an essential tool for observing and analyzing origin-destination (OD) data, which encodes flows between geographic locations, e.g., in applications concerning commuting, migration, and transport of goods. However, depicting OD data often encounter issues of cluttering and occlusion. To address these issues, many visual designs feature data abstraction and visual abstraction, such as node aggregation and edge bundling, resulting in information loss. The recent theoretical and empirical developments in visualization have substantiated the merits of such abstraction, while confirming that viewers' knowledge can alleviate the negative impact due to information loss. It is thus desirable to map out different ways of losing and adding information in origin-destination data visualization (ODDV). We therefore formulate a new design space of ODDV based on the categorization of informative operations on OD data in data abstraction and visual abstraction. We apply this design space to existing ODDV methods, outline strategies for exploring the design space, and suggest ideas for further exploration.

1. Introduction

Data visualization methods are useful to analyse flows between geographic locations, for instance commuting, migration, and transport of goods. The most common visualization technique is the flow map, which show the movements using edges or arcs on a map. Flow maps have a long history, starting in the 19th century with maps by Henry Drury Harness [Rob55] and Charles Joseph Minard [Rob67]. In the last decades many other techniques have been proposed, e.g., the OD map, (a grid of nested maps) [WDS10], and glyph-based visualization [AAFW17, YWZ*19].

Visualizing origin-destination (OD) data is far from trivial. A de facto standard flow map depicting a network over a geographical map often result in cluttering and occlusion for any large number of flows. There are various data processing algorithms for reducing cluttering and other visual encoding methods for alleviating occlusions. For example, Figure 1 shows three visual representations depicting measured OD data of commuting traffic flows among municipalities in the Netherlands [Sta21]. The generic visual encoding (top-left) exhibiting cluttering and occlusion. Filtering out some nodes and introducing colors (bottom left) can help but still make one wonder if it could be further improved.

Given a new visual design problem, designers of OD data visualization (ODDV) typically gain inspiration from the literature or rely on their prior knowledge. Analogically, this is like exploring unorganized data without a map. It is hence desirable to formulate a design space for ODDV, which can act as such a map, allowing designers to organize existing ODDV designs and explore new designs systematically.

We noticed that many ODDV methods involve removing and adding information, e.g., node filtering and edge coloring in the bottom-left of Figure 1, and geographical deformation and node duplication in [WDS10]. As information theory can explain the benefit of informative changes in visualization [CG16], we thus used different types of informative modification to OD data to structure a new design space. Using such a design space, we were able to discover a design option, as shown on the right of Figure 1, for improving the visual representations on the left of Figure 1. The new ODDV appears to "draw less information", e.g., by drawing each edge from only the half-way point to its destination and by summarizing the missing information about the outgoing flows as a small doughnut-chart at each origin. However, it allows viewers to perceive more information.

Most OD datasets are typically very large. The ODDV designers are constantly looking for means to draw less while preserving useful information. One novel feature of our design space is to categorize ways for "drawing less (and sometimes more) information". Analogically, like a map, a design space does not tell designers where to go, but enables them to find the best way as they have the knowledge about their data, users, and tasks.

2. Related Work

Origin-destination data visualization (ODDV) is an area of geospatial data visualization, which has a relatively long history and extensive literature. There are many wonderful books, such as those by Bertin [Ber83], MacEachren [Mac04], and Kraak and Ormeling [KO09], and literature surveys on cartography [Tob04, NK16]. Those particularly relevant to ODDV include surveys by Dodge et al. [DWL08], and Chen et al. [CGW15].

The de facto standard ODDV method is the flow map, with (directed) edges between origins and destinations. There are many variations for drawing the edges (e.g., straight lines vs. curved lines). Jenny et al. compared three pairs of design options [JSM*16]. Flow maps are effective for a small OD dataset. Otherwise, occlusion will be a major problem. A common solution is to group nodes and edges clusters (e.g., [LBR*16]). Edge routing techniques are introduced to enable semantically-meaningful clustering. While Minard handcrafted the routes of the edges, a few algorithmic methods have been proposed [PXY*05, VBS11, NB13].

When geospatial information is not the focus, an adjacency matrix (OD matrix) offers a compact visual representation of the flow network. Spatial clustering techniques have been used to order the rows (origins) and columns (destinations) [Guo07, LCvL17]. The OD map [WDS10, SKDW12, SKD14] provides matrix representations with some geospatial information by relating to the position of matrix cells coarsely, and often hierarchically, to the corresponding geographical locations.

Glyph-based visualization has been used to convey summary statistics about the flows related to individual locations. Glyphs can be placed on the exact geographic locations on a map [AAFW17] or spatially-deformed positions on a grid layout [YWZ*19]. When OD data features temporal changes, some ODDV methods use one spatial dimension for time. In a few designs, a 2D location is mapped radially on to an angular coordinate, while temporal data is depicted as small statistical charts placed along the polar axis (e.g., [DBS*11, LWLY15]). Many ODDV methods make use of different visual representations though multiple or composite views (e.g., [BBBL11, YDGM17]).

3. Origin-Destination Data Visualization (ODDV)

Origin-destination (OD) data describe the movements from origins to destinations. Many OD studies are about movements of humans, e.g., migration [Rob67, PXY*05, Tob87, SKDW12, SKD14, NB13, AS14, BBBL11, VBS11] and passenger transport [LWLY15, YWZ*19, LCvL17, LBR*16, AAFW17]. Other types of movements have also been studied, e.g., the export of goods [Rob67], the movement of animals [SPR*18], and the spreading of diseases [Guo07]. The focus of OD studies lies on the amount of flow that moves from an origin to a destination. The routes and times of the movements are of lesser importance in most OD studies.

A typical OD dataset is visualized in Figure 1. The dataset contains 11,577 commuting flows between 355 Dutch municipalities. For each pair of municipalities (A,B), the flow represents the number of people who commute from municipality A to municipality B. Strictly speaking these flow counts are numbers of employee jobs [Sta21], but for the sake of convenience we regard them as numbers of commuters henceforth.

It is worth noting that when one talks about OD data of countable flows, one usually refers to processed (i.e., aggregated) data where the raw data describes movements on individual level. The Dutch Commuting dataset is an example of such processed data, where the raw data consists of a record per commuter with coordinates for both home and work address. The processing steps include: grouping origins and destinations by municipalities, and for each pair of municipalities (A,B), counting the number of people who commute from A to B. Finally, in order to make sure individual persons cannot be identified by the data, edges with a very few number of commuters have been filtered out.

Figure 1(top-left) is an example of commonly used ODDV design. Origins and destinations have been drawn as dots on a base map. An edge between two dots represents a flow between them, where the line width encodes the amount of flow. The native method would be to draw straight arrows from origins to destinations. Since arrow heads would cause serious occlusion and each arrow would overlap its opposite arrow, edges are often drawn as curved lines without arrow heads.

In order to tackle occlusion, nodes and edges are often filtered or grouped. How this is done depends on the domain knowledge of the user and on the task at hand. In Figure 1(bottom-left), only flows with 500 commuters or more are depicted. In order to improve readability of the ODDV even more, the incoming flows for the four major Dutch cities (which are also municipalities) have been colored. With their knowledge, Dutch people can immediately recognize these cities.

Figure 1(right) is an example of a new ODDV design that was found by exploring the design space introduced in this paper. The edges are drawn from the half-way point to the destination, focusing on the incoming flows and preventing a significant amount of mutual occlusion of different color lines as in Figure 1(bottom-left). The outgoing flows are summarized by using small doughnut-charts, which are drawn instead of dots. The size of the charts represents the amount of outgoing flow, with a lower bound (in this case, 500). The colors indicate the proportion of people who commute to each of the four colored cities, with blue being used for people who commute to other municipalities and grey for people who work in their home municipality.

For the design of an ODDV, it is important to know both the user and task concerned. The ODDV shown in Figure 1(right) was designed for people who have topographical knowledge about the Netherlands. Typical use cases of such an ODDV include observing statistical patterns on commuting in the Netherlands and policy making regarding the infrastructure of roads and public transportation. An example of the latter use case is the following. When a municipality is located near a large city while the majority of people commute to another large city that is farther away, this may be an indication that the public transport facilities to the nearby city can be improved. The quality of an ODDV is reflected by whether such tasks can be easily executed.

4. Design Space of ODDV

Design space is multidimensional parametric space, which facilitate the registration of existing design options in an organized manner and enable designers to formulate new design options by systematically exploring different parametric combinations. In the context of visualization, a narrow interpretation of the term focuses on the parameters within the scope of visual representations. With a broad interpretation, one may include other parameters (e.g., variations of data, users, and tasks). While the broad interpretation may predicate an ideal, comprehensive design space, it will not be attainable easily. Many visualization researchers have constructed taxonomies, ontologies, and design spaces for different parts of the broad space, constituting a divide-and-conquer approach. Our work falls into the narrow interpretation of the term of design space.

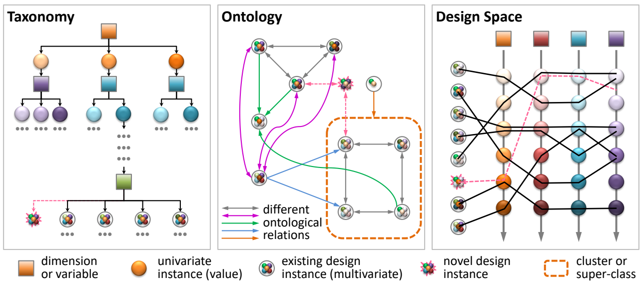

Figure 2: *Comparative illustration of taxonomy, ontology, and design space. A design space can aid the exploration of new designs.*

Taxonomies, ontologies, and design spaces can all serve the purpose of concept categorization. As illustrated in Figure 2, their structural differences entail their relative merits in supporting different design tasks. For example, a taxonomy is particularly effective for categorizing and searching for concepts (e.g., interaction techniques) in a hierarchical and scalable manner. An ontology records the relationships among different concepts (e.g., different visualization tasks in a group of workflows), facilitating the discovery of related concepts from one or more seed locations. In comparison, a design space enables the comparison of design options (or concepts in general) in their parameter space, allowing easy identification of the missing design options.

Since OD datasets are typically large, many ODDV designs are focused on information loss, e.g. by aggregation. Sometimes adding previously-unavailable information may also be applied, e.g. by taking into account routes. If one assumes that the added information has been available, adding implies reintroducing lost information selectively, i.e., losing less information. Naturally we consider the parameters related to information processing can be used to organize an ODDV design space. According to the latest information-theoretic explanation about visualization processes, information loss is a ubiquitous feature of visualization [CG16]. This provides us with the theoretical underpinning to organise a design space based on how information is lost or reintroduced into ODDV. We detail this theoretical background in Appendix A.

Some design spaces in visualization literature for other domains are based on an abstraction of tasks [ENXS20, LJS20]. However, this is not easy since the choice of design options does not only depend on the task, but also on the user and the data. For instance, an ODDV designer may ask a question: what information that is contained in the data is needed for the task at hand taking into account the domain knowledge of the user? We hence focus on constructing a design space for design options. Instead of prescribing decision options based on tasks, we assume that as long as ODDV designers have gained adequate understanding of data, users and tasks, they are able to judge if a design option is appropriate.

We will start with a formal definition of the OD data structure. This is followed by four subsections detailing the four dimensions of the proposed design space. The first two dimensions are about transforming the graph structure of the OD data, with dimension 1 for the node set and dimension 2 for the edge set. Data processing is mostly associated with these two dimensions. The next two dimensions are about individual elements of the graph, with dimension 3 for nodes, and dimension 4 for edges. Visual mapping is mostly associated with these two dimensions.

Information theory forms the basis of our design space. When data is depicted and perceived visually, a certain amount of information will be lost, even though a user may not be aware of it. Our aim is to make explicit which parts of the information are lost and what the consequences are for the visualization. This will allow ODDV designers to make an informed choice in conjunction with their knowledge of the users and the tasks at hand.

4.1. OD Data Representation

The basic structure of an OD dataset is an enriched graph structure

$$G = (U, E),$$

where $U = (u_1, u_2, \dots, u_n)$ denotes the set of nodes (n in total) and $E = (e_1, e_2, \dots, e_k)$ denotes the set of edges (k in total).

Because of its geospatial context, each node $u \in U$ has geographical coordinates (x, y) and optional attributes $\psi_1, \psi_2, \dots$:

$$u = (x, y, \psi_1, \psi_2, \dots)$$

Each edge $e \in E$ has a starting-point (origin) $u \in U$, an endpoint $v \in U$, and optional attributes $\xi_1, \xi_2, \dots$:

$$e = (u, v, \xi_1, \xi_2, \dots)$$

Given an OD dataset, an ODDV process transforms its data representation to a visual representation through a series of algorithmic steps. In abstraction, these algorithmic steps are referred to as transformation functions. Traditionally, these functions may be labelled as “filtering”, “abstracting”, “visual mapping” and other terms by considering the ODDV process as a pipeline. In many ODDV implementations, some of these labelled steps are often closely integrated rather than being neatly separated into preprocessing and visual encoding. For example, one may consider moving nodes and edges using a forced-direct layout algorithms as either preprocessing or visual encoding.

We therefore introduce a data-centric classification scheme of these functions for the design space, which focuses on what kinds of data in $G = (U, E)$ being transformed. Based on the above definition, we can observe four primary types of transformations, i.e., those for a node set U, those for an edge set E, those for a node u, and those for an edge e. Any complicated transformation can be decomposed into primitive functions belonging to a single family. Considering each primary type as a dimension of the design space, we can map out a specific ODDV design into these four dimensions. In the following two subsections, we examine the two dimensions for U and E first. After giving some more definitions, we examine another two dimensions for u and e.

4.2. Dimension 1: Transformation of a Node Set

For this dimension, we consider the transformation of the set of nodes. Formally, let $\mathcal{U}$ be the set of all possible node sets. This dimension consists of a variety of transformation functions in the form of $F_1 : \mathcal{U} \to \mathcal{U}$, such that

$$U' = \{u'_1, u'_2, \dots, u'_{n'}\} = F_1(\{u_1, u_2, \dots, u_n\}) = F_1(U)$$

where $U, U' \in \mathcal{U}$. We consider the four basic transformation functions filter, group, add, and split, which can be used in combination:

Filter This transformation function filters nodes that meet specified conditions. Two commonly used conditions are whether the nodes are located in a specified geographic area and whether the values of an numeric attribute are higher than a specified threshold value (Figure 3(a) and (b) respectively).

Group This transformation function groups the nodes. Each group is replaced by a new node. Four commonly used grouping functions are depicted in Figure 3: group by grid cell (e), administrative area (f), cluster (g), and by same coordinates (h).

Add This transformation function adds new nodes. It is only commonly used in combination with a group function to represent empty groups. For instance, when the nodes are grouped by grid cell as shown in Figure 3(e), additional nodes may be added for empty grid cells (Figure 3(c)).

Split This transformation function splits a node into multiple new nodes. This function is not commonly used, but may be useful when nodes have certain weights, where the purpose of the ODDV is to show equally weighted nodes. In that case, the nodes can be split by weight (see Figure 3(d)).

Note that the transformation function F_1 does not necessarily have to be applied before F_2, the transformation function of the edges which we will describe in the next section. For instance, when edges are grouped, the end points of those edges are automatically grouped as well. Note that the original end points can still be added as new nodes.

4.3. Dimension 2: Transformation of an Edge Set

The transformation of the edge set is similar as the transformation of the node set. Let $\mathcal{E}$ be the set of all possible edge sets. Then $F_2 : \mathcal{E} \to \mathcal{E}$ described the transformation function such that

$$E' = \{e'_1, e'_2, \dots, e'_{l'}\} = F_2(\{e_1, e_2, \dots, e_l\}) = F_2(E)$$

where $E, E' \in \mathcal{E}$. Similar to node transformation functions, edge transformation functions can be categorized as one of the following basic functions, or a combination of them.

Filter Edges are typically filtered by core visual variables, for instance length or attribute value (Figure 4(a) and (b)).

Group Edges are commonly grouped by core variables, such as angle of direction or distance (Figure 4(e) and (f)). It is also possible to group edges that have similar end points (Figure 4(e)), which is automatically followed by an F_1 transformation of grouping nodes by same coordinates (Figure 3(h)).

Add Edges may be added in order to create a *complete graph*, which is a graph that contains an edge between any two nodes in both directions, as illustrated in Figure 4(c). This is especially useful for OD matrices, where the rows correspond to origin

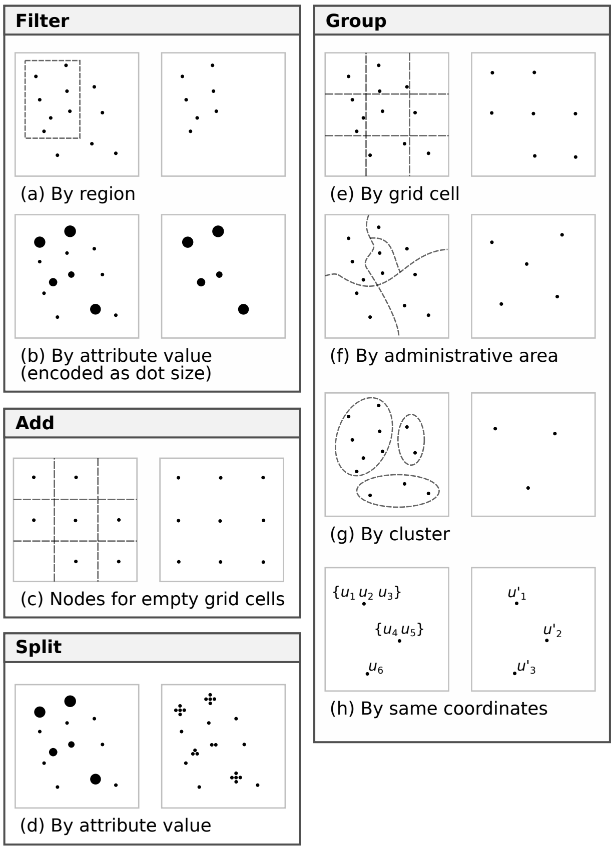

Figure 3: *F_1 transformation functions of the node set. Per type (fill, group, add, and split), one or more examples are shown. For each pair of diagrams, the left-hand side depicts the original node set, and the right-hand side the transformed node set.*

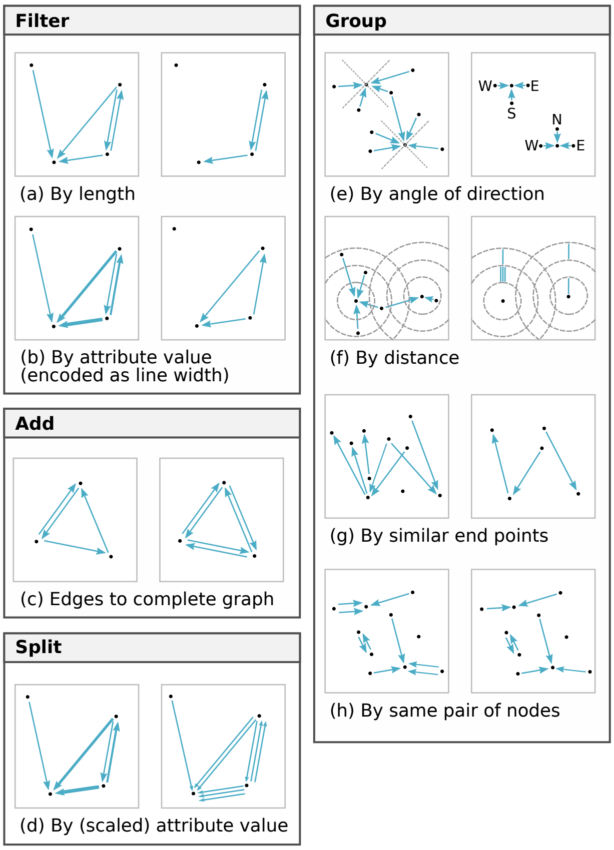

Figure 4: *F_2 transformation functions of the edge set. Per type (fill, group, add, and split) one or more examples are shown. For each pair of diagrams, the left-hand side depicts the original edge set, and the right-hand side the transformed edge set.*

nodes, the columns to destination nodes and each cell to a link; a complete graph is needed to have data points for each cell.

Split Splitting an edge set is not a common operation. It might be useful in cases where the edges have certain weights (Figure 4(d)).

4.4. Core, Non-core, and Derived Variables

Before we examine the next two families of transformation functions F_3 and F_4, we introduce a few more definitions that will help the categorization of F_3 and F_4. The *data variables* in an individual node record u or edge record e fall into three categories:

- *Core data variables*: These variables are almost ubiquitous to the data representations in almost all ODDV applications, and they are assumed to be explicitly defined in the source data entering a ODDV process. The core variables of a node are its two coordinates x and y. The core variables of an edge are two ordered nodes u and v and their associated coordinates $(x,y)_u$ and $(x,y)_v$.
- *Non-core data variables*: These variables are commonly referred to as attributes, and are denoted the records of u and e using Greek letters ψ and ξ. They are optional and application-dependent. They are explicitly defined in the source data.
- *Derived data variables*: These variables are not explicitly defined in the source data, but can be derived from those explicitly-defined variables. For example, giving the order of u and v, and their coordinates $(x,y)_u$ and $(x,y)_v$, we can calculate the direction $(\Delta x, \Delta y)$, the middle point (c_x, c_y), the distance $\text{dist}(u,v)$, a straight line segment (i.e., all points between u and v), and so on.

When a node is visualized, its variables, core, non-core, and derived, are encoded as *visual variables* depicted by different parts of the visual object corresponding to the node, including its location, size, shape, colour, etc. Similarly when an edge is visualised, its variables are encoded as visual variables, including the locations of its two nodes, a line between two nodes, an arrow on the line, etc.

The core visual variables of a node (or edge) are all visual variables that represent the core variables of a node (or edge) or derived variables that are derived only from core variables. For an edge, for example, visual variables for depicting $(x,y)_u$, $(x,y)_v$, direction, length are all core visual variables, but those for flow capacity and flow capacity / length are not core visual variables.

A visual variable may be *explicitly* or *implicitly* encoded. For example, when an edge depicted by a line between two dots, the length is explicitly encoded, the mid-point is implicitly encoded,

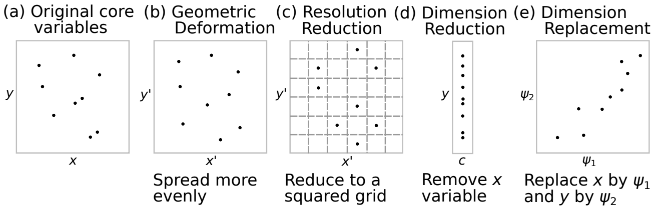

Figure 5: *Examples of F_3 transformations: (a) the original nodes, and (b)-(e) results of the four types of transformations.*

and the ordering-direction is not encoded. When a large hollow or solid circle is used to depict a node, its (x,y) coordinates are implicitly encoded.

4.5. Dimension 3: Transformation of Individual Nodes

Let $\mathbb{u}$ be the set of all possible node records. The transformation of an individual node $u \in \mathbb{u}$ is a function $F_3 : \mathbb{u} \rightarrow \mathbb{u}$ such that

$$u' = (x', y', \psi_1', \psi_2', \ldots) = F_3\big((x, y, \psi_1, \psi_2, \ldots)\big) = F_3(u)$$

A transformation, such as filtering, grouping, adding, and splitting, is applied to the data variables within the record of u. It is relative easy to imagine how such a transformation may be applied to those non-core data variables. For example, one may delete ψ_3, add ψ_6', group ψ_3 and ψ_4 as $\psi_7' = \psi_3 + \psi_4$, or split ψ_5 into its quotient $\psi_8' = \lfloor \psi_5 \div 10 \rfloor$) and remainder $\psi_9' = \psi_5 \mod 10$. However, it may be less obvious when such transformations are applied to the core data variables x and y. We therefore categorize such transformations in terms of what has been changed, with an additional note on how.

Geometric Deformation. Move (x,y) to (x',y') according to algorithmic decisions as illustrated in Figure 5(b). Change its core visual variables. This is a combined action of **filtering** and **adding**.

Resolution Reduction. Reduce the resolution of x or y or both (Figure 5(c)). Change one or more core visual variables. Visually, this is a special case of geometric deformation. This is a **grouping** action in terms of the variable(s) concerned.

Dimension Reduction. Remove one of the core variable, x or y. Visually, this is a special case of the above two, as removing a variable is an effect to reduce its resolution to single value. For example, if x is removed, a node originally at (x,y) will now be displayed at (c,y) on a 2D screen, where is c is a constant (Figure 5(d)). Therefore, $x' = c$. This is a **filtering** action in terms of the removed data variable and **grouping** in terms of mapping all x values to c.

Dimension Replacement. Replace one or more core variables with non-core variables or derived variables, and encode the latter as if they are core. e.g., replace x with ψ_1 and/or y with ψ_2 (Figure 5(e)). This is a combined action of **filtering** and **adding**.

However, unlike most subjects (e.g., communication, compression) underpinned by information theory, in visualization, the encoder (i.e., data processing, visual mapping and visual display) and decoder (i.e., viewing, perception, and cognition) are not engineered as a constituent pair by the same developer [CJ10]. What is being encoded is not assured to be decoded. As illustrated in Figure 6, different visual representations of a node may capture a viewer's attention differently, may demand different cognitive load to retrieve its (x,y) values. As the above four categories cannot capture the non-binary states of the decoder, we introduce two additional categories **Attenuation** and **Enhancement**.

These two categories need a common reference state, which is the *de facto* standard encoding of a node. We first define visual variables x and y to be the *de facto* standard set referred to as the *node norm* (NN). In principle, it is not absolutely necessary to display data variables (x,y) as (x,y) explicitly. One could, for instance, display them as $(x,-y)$, $(x,\log_{10} y)$, or $(x, x-y)$, where the value of y can be inferred from what is displayed. As this may be uncommon or less intuitive for nodes, we will discuss this kind of visual encoding for edges in detail and the notion of implicit encoding will become clear in Section 4.6.

Once we have the node norm, we can establish **Normal Encoding of NN** as a category of F_3, and the reference point for attenuation and enhancement. We define the most conventional visual encoding of a node shown in Figure 6(a) as the normal encoding. In (b), a reduction of the contrast between its color and the background makes it less noticeable, hence attenuating its presence. Meanwhile, in (c), displaying the node using flicking animation makes it more noticeable, hence enhancing its presence. In (d), when encoding the node with a larger circle, the node presence is enhanced, but the perception of the (x,y) values is attenuated. In (e), when a heart icon is used in stead of the black dot in (a), the x value can be perceived more or less similarly, but the y-value is rather uncertain. The perception of (x,y) values can be enhanced using the cross in (f) or the tracing lines (typically activated interactively) in (g). In the following discussion, we consider the term x- or y-dimension include both the presence, notability, and the value of a dimension. The three additional categories are summarised as:

Normal Encoding. Show core visual variables (x,y) of a node using the conventional visual representation, which entails a black dot on a white background or vice versa. The size of the dot is expected be defined differently for difference media (e.g., handheld, desktop, projection screen, etc.). We hence define the size-norm as the smallest size such that all target viewers can effortless identify it in an uncluttered layout. This is illustrated in Figure 6(a).

Dimension Attenuation. Show either or both of the core visual variables (x,y) in a way that causes slowing-down viewing or less accurate perception of the variable(s) concerned in comparison with the node norm. It is a **recoverable filtering** action in relation to the node norm as the information is encoded, but may not be more difficult to decode than the node norm.

Dimension Enhancement. Show either or both of the core visual variables (x,y) in a way that enables speed-up viewing or more precise perception of the variable(s) concerned. It is a kind of **adding** action in relation to the node norm.

4.6. Dimension 4: Transformation of Individual Edges

Similarly, let $\mathbb{e}$ be the set of all possible edge records. The transformation of an edge $e \in \mathbb{e}$ is a function $F_4 : \mathbb{e} \rightarrow \mathbb{e}$ such that

$$e' = (c_1, c_2, \ldots, \xi_1', \xi_2', \ldots) = F_4\big((u, v, \xi_1, \xi_2, \ldots)\big) = F_4(e)$$

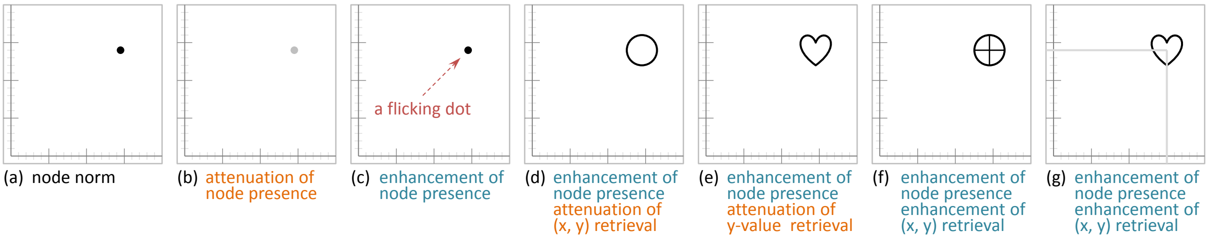

Figure 6: *Examples of different visual representations of nodes in ODDV. Although the core variables (x,y) of the node in each case are encoded, their decoding may demand different levels of cognitive load and may incur different amounts of uncertainty.*

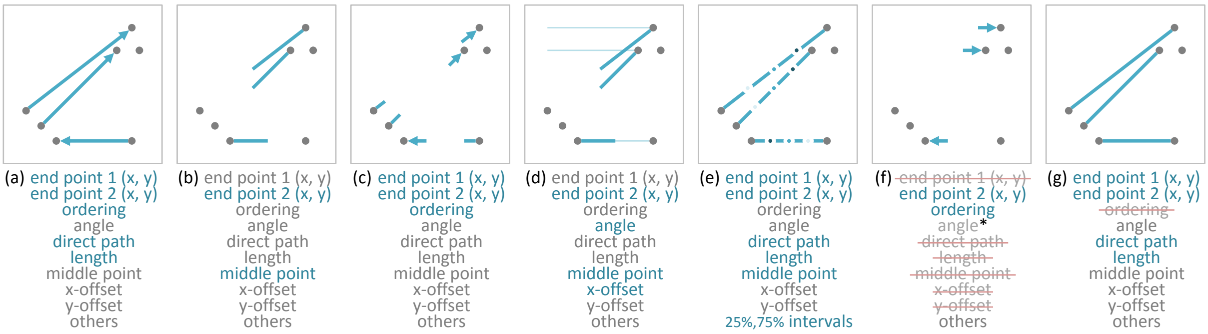

Figure 7: *Examples of different visual representations of edges in ODDV. A list of core visual variables are listed below each example, where explicitly-encoded ones are labelled in cyan, implicitly-encoded ones in grey, and un-encoded ones in grey with a strike-through line. (*) Note that in (f) the angle is implicitly encoded, but with significant resolution reduction as one can still differentiate leftward from rightward.*

where c_i is an original core variable or derived core variable, such as x- or y-position, Δx, Δy, length, angle, etc.

For a node record u, x' and y' are core data variables as well as core visual variables. For an edge record, e, the relationship between the two types of variables is much more complicated. We rarely just draw two end-points. As shown in Figure 7(a), we draw an arrow to indicate the ordering of the two end-points, and a straight line to depict the direct path from one to another. In fact, neither the arrow is essential as depicted in the second example, nor the straight line as in the third example. This is because a viewer can infer such implicitly-encoded visual variables from those explicitly-encoded variables, except it may take more time, incur more cognitive load, or cause more errors in visualization.

As the example shown in Figure 7(a) is the most commonly-used visual representation for an directed edge, we consider the seven core visual variables explicitly encoded there as a *de facto* standard set. We call this set as an *edge-norm* (EN), which includes the (x,y) positions of the two end-points, their ordering, the direct path, and its length. There are other core visual variables that are implicitly shown. The consideration of explicitness and implicitness makes the categorization of F_4 much more complicated than F_3. With F_3, if the x-dimension is not displayed, one normally cannot infer it from y. As exemplified in Figure 7, it is not the same with edges because of information redundancy in EN.

In addition, similar to a node, the perception the core variables of an edge can also be attenuated or enhanced. Following the same reasoning about the examples in Figure 6, we can easily reason the attenuation and enhancement in Figure 7. In general, a change from explicit to implicit depiction usually causes an attenuation of perception. A change from implicit to explicit depiction usually leads to an enhancement.

Similar to the node record, it is relatively easy to define and exemplify filtering, grouping, adding, and splitting. For the core visual variables $c_1, c_2, \ldots$, we assume that the transformation F_4 will visually encode them explicitly. Using the EN set as the reference set, we can define filtering, grouping, adding, and splitting. For instance, in the second example in Figure 7, the (x,y) position of end-point 1, the direct path, and the length are removed by F_4 from the EN reference. They are filtered out by F_4 but are perceptually recoverable by a viewer. Meanwhile, the middle point of each line is added by F_4 as an extra core visual variable. These middle points help recover the information about three variables filtered out. In the last example, arrows are removed by F_4. The ordering information is filtered out and it is perceptually irrecoverable. Now we can categorize the transformations of these visual variables in terms of what has been changed to the EN, with an additional note on how.

Normal Encoding. Show core visual variables in EN explicitly and show all other core visual variables implicitly. This is illustrated in Figure 7(a).

Resolution Reduction. Show some core visual variables in EN with reduced resolutions. This is a **grouping** action as edges with different data records may visually appear to be the same. As illustrated in Figure 7(f) with respect to angle, the angles are grouped into two groups: from west to east and from east to west.

Dimension Attenuation. Show some core visual variables in EN implicitly or in a way that cause slowing-down viewing or less accurate perception of the variables concerned in comparison with the normal encoding of EN. It is a **recoverable filtering**

action in relation to EN. This is illustrated in Figure 7: in (b) and (d) with respect to end point 1, direct path and length; in (c), to direct path and length; and in (d), to direct path.

Dimension Enhancement. Show some core visual variables not in EN explicitly or show some core visual variables in EN in a way that enables speed-up viewing or more precise perception of the variables concerned. It is a kind of **adding** action in relation to EN. Examples in Figure 7 are: (b) with respect to middle point, (d) with respect to middle point, x-offset, and angle, and (e) with respect to middle point and 25/75 intervals.

Dimension Reduction. Completely remove some core visual variables in EN. These core visual variables are no longer encoded, i.e., neither explicitly nor implicitly and thus no longer perceivable. It is an **irrecoverable filtering** action in relation to EN. In Figure 7, this is illustrated in (f) with respect to end point 1, direct path, and length, and in (g) with respect to ordering.

Dimension Replacement. Replace some core visual variables in EN with one or more non-core visual variables. This is a combined **filtering** and **adding** function in terms of core. In relation to EN, the removed core visual variables may or may not be recoverable. An example is Figure 7(b) where the length can be used to encode an attribute variable.

5. Application

The presented design space can help ODDV designers to see the big picture. Analog to using a map of a partly unknown territory, the design space can be used to find out where the existing points of interests, i.e. existing ODDV methods, are located and which part of the design space is unexplored. Another analogy with using a map is that it not only tells us where points of interests are, but also provides a better context of the locations.

In this section, we first add points of interests by classifying some existing ODDV techniques in the literature. We then outline several actions for exploring our design space. This is followed by a test case of using the design space to discover the ODDV for the Dutch Commuting dataset. Finally, we provide some ideas for further exploration of undiscovered parts in the design space.

5.1. Classification of ODDV Literature

In this subsection, we demonstrate how eight existing ODDV methods can be placed in our design space. The example visual designs of these methods are shown in Figures 10–17 in Appendix B, where we used zoomed-in views to annotate the nodes and edges.

In order to be complete, we describe the transformation functions starting from raw OD data. Since all examples describe mobility or migration of persons, we can safely assume that a raw OD dataset consists of a record per person, with the location of both origin and destination. As mentioned in Section 3, ODDV designers may not always have had access to these raw datasets. If that is the case, the transformation functions F_1 and F_2 (or part of them) have not been applied by the ODDV designers, but in the preprocessing stage.

All eight examples mentioned in Appendix B, except Figure 17, have similar transformations functions in Dimensions 1 and 2, namely group nodes, mostly by administrative area, and group edges accordingly. For the ODDVs that are based on the OD matrix (Figures 13, 14, and 15), edges have been add to complete the graph (Figure 4(c)).

The flow diagrams, shown in Figure 17, have more complex transformation functions in Dimensions 1 and 2. First, the nodes are grouped by cluster (F_1). Then, for each origin node, the edges are grouped by angle (in eight cardinal directions) and length (in three classes), which is a F_2 transformation. After that, the destination nodes are grouped accordingly, which is an F_1 transformation.

None of the eight examples have applied the node norm. The standard flow maps shown in Figures 10, 11, and 12 show the coordinates (x, y), even though some F_3 transformation functions have been applied. The other examples either have reduced or replaced the dimensions. Notice that the main difference between an OD matrix Figure 13 and an OD map Figure 14 is the third dimension; in the OD matrix the coordinates have been replaced by order in the matrix whereas in the OD map, the coordinates have been deformed to a nested grid, where origin nodes are shown in the large grid and destination nodes in the small grids.

None of the eight examples have applied the edge norm either. Although the examples in Figures 10, 11, 12, and 16 still show edges explicitly, all of them have applied some of the transformation functions mentioned in Section 4.6. The edges shown in the Figure 17 are different in the sense that they are part of the glyph design. The length of those edges does not represent distance, but flow (a Dimension Replacement).

The classification results not only provided insights into the underlying design choices, but also indicated which parts of the design space are more populated with ODDV methods than other parts. It is worthwhile for VIS researchers to examine those empty parts and explore new ODDV designs that would reside there.

In the remainder of this section, we first describe a test case for exemplifying how the design space in 4 may be exploited, and then discuss those less exploited areas.

5.2. Design Space Exploration

Like map exploration, the effectiveness and efficiency in navigating in the design space depends on the designers' knowledge of existing ODDV methods and their categorization in the four dimensions. Obtaining a good visual design is a balancing act that maximizes the positive impact of information loss while minimizes the negative impact and cost (see Appendix A). Here we outline several actions that may help deliver such a design.

Know the users. Users' knowledge about the data and application context and their previous experience of viewing similar data using the visual design can alleviate the negative impact of information loss. The choice of information type and reduction should be based on users' knowledge.

Know the tasks. Losing a type or amount of information may cause poor performance for some tasks, but may improve task performance for other tasks. The choice of information type and reduction should be based on users' tasks.

Benefits of interaction. When abstraction such as filtering and grouping is applied, interaction (e.g., adding and splitting) allows users to reintroduce some of the lost information. One can estimate the net benefit of such interaction, enabling a balancing act between information loss and the cost of information reintroduction.

Trade-off between dimensions. When one reduces information in one dimension, one may preserve part of the lost information in another dimension. For example, one may remove some nodes using grouping (F_1), add the group size as an attribute of the super-node, and visually encode the attribute (F_3).

Trade-off between explicit and implicit encoding. One may display edges using different combinations of explicit and implicit encoding of core visual variables. Likely, there are novel visual encodings to be discovered.

Start with simple information reduction. With a large OD dataset, cluttering is always an issue. One can start with information reduction by considering different categories of F_1, F_2, F_3, and F_4 transformations. If there is too much information loss, one can consider interaction and the above trade-offs as balancing acts.

Start with an existing design. Alternatively, one can start with an existing ODDV method, and use the design space to explore different trade-offs.

5.3. Test Case: Commuting in the Netherlands

In addition to using the ODDV design space to categorize various design ideas featured in the literature, we can also use the design space to explore new design ideas. In particular, because it is uncommon to have a design space structured based on the informative changes, it can provide opportunities for uncommon design ideas. After completing the specification of the design space in Section 4, we used the Dutch Commuting dataset described in Section 3 and the two initial ODDVs on the left of Figure 1 to test and explore the design space [Sta21].

We noticed that the visual encoding on the bottom-left has already featured an F_2 transformation for filtering out some edges, and an F_4 transformation for enhancing the presence of certain edges. Interestingly, color-coding edges based on an attribute of its nodes (e.g., population of a city) does not add or remove any core visual variables in the edge norm, but falls into the category of Dimension Enhancement as it makes some visual variables more perceivable (e.g., end-points, direct path, and length). In information theory, it is related to automated error detection and correction by the decoder as viewing the edges is rather "noisy" (see [CJ10] for a detailed explanation).

Meanwhile, the missing arrow heads indicate an F_4 transformation, i.e., Dimension Attenuation. The ordering can still be perceived by the curve angle, albeit much harder than via arrows; edges are bend to the right-hand side from the origin point of view.

Following a quick scan of the figures in Section 4, we noticed that the half-edge encoding in Figure 7(b) could help address the cluttering issue while bringing back the ordering information. We were aware that the distance between the two end-points of an edge could be perceived much shorter than in reality and that the dimension of the origin nodes might be attenuated too much. We considered to use glyphs as shown in Figure 17, but the original glyph design unfortunately clashed with the half-edge encoding. With another scan of the figures in Section 4, we noticed that a glyph based on Figure 6(d) or (f) could avoid a clash with the half-edge encoding. This led to the ideas of using a small doughnut chart or pie chart. After some prototyping experiments, we narrowed the design down to the option using small doughnut charts. The design in Figure 1(right) has been implemented and deployed as a web-based interactive map by Statistics Netherlands [Sta21].

A post-hoc analysis helped us notice some other merits. Since an edge meets its opposite edge at their half-way, one can connect the two end points perceptually, and moreover, compare the volumes between both edges. This is generally easier for flows between color-coded cities than for other flows. Another benefit of our design is that the small doughnut chart enables us to show additional information. We opted for a summary of outgoing commutes, which was not possible with the two initial ODDVs on the left of Figure 1 for small municipalities.

5.4. Scope for Further Exploration

Figures 3, 4, 5, 6, and 7 in Section 4 are not exhaustive lists of instances of design options. They are abstract illustrations that can be used to prompt VIS researchers to instantiate design ideas suitable for the data, users, and tasks that they know well. We hope that with a community effort, these sets of abstract instances will be extended in the future. Here we focus on the four dimensions of the design space, and provide our observations as to aspects in the design space that may offer interesting, and potentially rewarding, areas for exploration. We use the symbol ► to indicate such an area.

Dimension 1: Transformation of a Node Set. As discussed in Sections 4.2 and B, the four types of transformation functions applied to a node set (i.e., filter, group, add, and split) are typically implemented as part of interactive visualization, such as zooming and abstraction. Much of the existing work in the literature has focused on analytical algorithms as well as human-computer interaction. Because there is an informative trade-off (see Appendix A), the decisions as to what to filter and add and how to group or split are mostly data-, user-, and task-dependent. The exploration for new application-specific algorithms for these transformations will continue, as well as for new interaction modalities.

► What we can explore more is perhaps the "resolution" of filtering. When a node is filtered out, the typical consequence is that the node and its edges are no longer available to the visual mapping. When a number of nodes are grouped together, their edges are also grouped together. Hence the decisions about whether to keep a node or not in the node set are of a binary nature. In order to explore the intermediate states between the binary decisions, one could explore different functions F_3 and F_4 for those nodes to be transformed. For example, for a node $u \in U$, a filtering transformation $F_1(U)$ may not simply decide if it remains in U' or not. Instead, it may "downgrade" a node to a "lower" state. Here the word "lower" implies that a transformation F_3 (i.e., in Dimension 3 of the design space) may make use less information bandwidth to encode such a node visually, e.g., resolution reduction, dimension attenuation, or dimension reduction.

► In general, there are more reports on filtering and grouping, but much less on adding and splitting. In many application scenarios, there are needs for "add" transformations (e.g., dynamic data streaming) and "split" transformations (e.g., ungrouping different modes of transport). These transformations that introduce more information often cause problems to other dimensions, e.g., Geometric Deformation, Resolution Reduction, and Dimension Reduction in F_3. These problems pose challenges as well as opportunities for new innovations.

Dimension 2: Transformation of an Edge Set. Similar to the discussion in the above section, transformation functions in this dimension of the design space typically make binary decisions about an edge $e \in E$. The information about e is either entirely available or fully lost. Since an optimal visual design reflects the trade-off among alphabet compression, potential distortion, and cost, allowing for different amount information loss can enable designers to explore more design options.

► One area of exploration is to introduce some intermediate states, between the binary decisions of either entirely available or fully lost, for each edge in an edge set. After a "filter" or "group" transformation, an edge e may be "downgraded", and an F_4 transformation F_4 may use less information bandwidth to encode the edge visually.

► Similar to Dimension 1, new research effort on "add" and "split" transformations will be much appreciated.

Dimension 3: Transformation of Individual Nodes. ► The concept of "node norm" is new. It raises a research question as to how different ways of encoding nodes affect their perception. Figure 6 shows that the most conventional ways of encoding a node is not optimal for perceiving the values of (x,y). In other words, most ODDVs would cause a non-trivial amount of perception errors if perceiving (x,y) is a visualization task. On the other hand, if perceiving (x,y) is not a visualization task, why it not always better to use visual representations with limited geospatial information, such as the ODDVs in Figures 13, 14, 15, and 16. This leads to a more fundamental question as to how viewers benefit from displaying a node at (x,y).

► Many existing visual designs, e.g., the nodes in Figures 13, 14, 15, and 16, feature transformations that fall into the categories of Geometric Deformation, Resolution Reduction, Dimension Reduction, and Dimension Replacement. Some common rationales includes "requirements by domain experts" and "task-dependent designs". There are scopes for gaining deeper understanding about behind these rationales, such as what are the potential demerits, what tasks may suffer from such demerits and what may not, and what human knowledge may alleviate the impact of demerits.

► Encoding nodes with colors, shapes, sizes, etc. is often reasoned on the basis of showing some application-specific attributes (e.g., population). It is less common to reason the positive or negative impact of such visual encoding on the perception of core visual variables of nodes (x,y). Since the positive or negative impact is rarely avoidable, it is desirable to examine and understand such impact in a systematic or organized manner.

Dimension 4: Transformation of Individual Edges. ► In comparison with "node norm", the concept of "edge norm" is much more interesting. While a simple directed edge seems to convey many core visual variables related to an edge, there are numerous alternative visual representations can do almost the same. From a purely mathematical perspective, it may not appear to be an issue as many these "additional" visual variables in the edge norm are not independent. However, saying that one variable can be derived from others differs significantly from saying it can perceived just like others. This topic will likely provide a fertile ground for further research, including theoretical development, innovative designs, and empirical studies.

► Many existing visual designs feature transformations that fall into the categories of Resolution Reduction, Dimension Reduction, and Dimension Replacement. Similar to Dimension 3, there is a need to gain deeper understanding about behind these rationales.

► Encoding edges with colors, thickness, shapes, etc. is often reasoned on the basis of showing some application-specific attributes (e.g., transport mode and traffic volume). Similar to Dimension 3, it is desirable to examine and understand the impact of such visual encoding on the core visual variables in the edge norm in a systematic or organized manner.

6. Conclusions

We have introduced a 4D design space for ODDV methods with the purpose to gain insights about the visualization of OD data, and provide a means for exploring potentially novel designs systematically. Some years ago, it might be unthinkable to use information loss as the central theme to categorize ODDV methods because such loss would only be viewed negatively. With support from the mathematical reasoning about the merits and demerits of information loss in [CG16], we have found that the categorization scheme based on different types of informative changes is rather appropriate for ODDV due to the facts that ODDV is useful and information loss is ubiquitous in ODDV.

OD data is typically large. With the design space, ODDV designers can now pursue a design process by focusing on **which** information to keep, loose, attenuate, enhance, and add, in conjunction with their knowledge of about the user and the task at hand. The designers can then scan the design space as if it were a map about various transformations in abstraction, and identify **how** to keep, loose, attenuate, enhance, or add information, in conjunction with their creativity and experience. The design space itself cannot replace designers' knowledge about users and tasks and their creativity and experience in design, but can help structure the design process more effectively and efficiently. With the visualization of the Dutch Commuting dataset in Figure 1(right), we have demonstrated how this can be done.

We hope to engage the VIS community to continue design space research, formulating general methodologies for design space exploration. For those aspects identified in Section 5.4, it will need a substantial amount of effort to fill in these gaps. In addition, there are no doubt many other aspects to be identified. Furthermore, the design space is not a static fixture, and can and will be improved or extended by future work, e.g., for hypergraphs. We also hope that we have shown the merits of information theory for data visualization. We highly recommend to apply information theory to other data types and their visual representations.

Acknowledgement

This work has been made possible by the Network of European Data Scientists (NeEDS), a Research and Innovation Staff Exchange (RISE) project under the Marie Skłodowska-Curie Program. We would like to express our gratitude to the people who facilitated this project, in particular Dolores Romero Morales from Copenhagen Business School. Also thanks to Sofie de Broe from Statistics Netherlands for the opportunity to participate in this project.

References

[AAFW17] ANDRIENKO G., ANDRIENKO N., FUCHS G., WOOD J.: Revealing patterns and trends of mass mobility through spatial and temporal abstraction of origin-destination movement data. *IEEE Transactions on Visualization and Computer Graphics 23*, 9 (October 2017). doi:10.1109/TVCG.2016.2616404. 2, 17

[AS14] ABEL G. J., SANDER N.: Quantifying global international migration flows. *Science 343*, 6178 (2014), 1520–1522. 2, 17

[BBBL11] BOYANDIN I., BERTINI E., BAK P., LALANNE D.: Flowstrates: An Approach for Visual Exploration of Temporal Origin-Destination Data. *Computer Graphics Forum* (2011). doi:10.1111/j.1467-8659.2011.01946.x. 2

[Ber83] BERTIN J.: *Semiology of Graphics*. University of Wisconsin Press, 1983. 2

[CARSS21] CHEN M., ABDUL-RAHMAN A., SILVER D., SBERT M.: A bounded measure for estimating the benefit of visualization: Case studies and empirical evaluation. arxiv 2103.02502, 2021. 16

[CE19] CHEN M., EBERT D. S.: An ontological framework for supporting the design and evaluation of visual analytics systems. *Computer Graphics Forum 38*, 3 (2019), 131–144. 15

[CG16] CHEN M., GOLAN A.: What may visualization processes optimize? *IEEE Transactions on Visualization and Computer Graphics 22*, 12 (2016), 2619–2632. 2, 3, 10, 13, 15

[cGS*19] ÇÖLTEKIN A., GRIFFIN A. L., SLINGSBY A., ROBINSON A. C., CHRISTOPHE S., RAUTENBACH V., CHEN M., PETTITAND C., KLIPPEL A.: Geospatial information visualizationand extended reality displays. In *Manual of Digital Earth*, Guo H., Goodchild M. F., Annoni A., (Eds.). Springer, New York, 2019, ch. 7, pp. 229–277. 16

[CGW15] CHEN W., GUO F., WANG F.-Y.: A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems 16*, 6 (2015), 2970 – 2984. 2

[Che21] CHEN M.: A short introduction to information-theoretic cost-benefit analysis. arXiv:2103.15113, 2021. 13

[CJ10] CHEN M., JÄNICKE H.: An information-theoretic framework for visualization. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1206–1215. 6, 9

[CS21] CHEN M., SBERT M.: A bounded measure for estimating the benefit of visualization: Theoretical discourse and conceptual evaluation. arxiv 2103.02505, 2021. 16

[CT06] COVER T. M., THOMAS J. A.: *Elements of Information Theory*. John Wiley & Sons, 2006. 13, 15

[DBS*11] DROCOURT Y., BORGO R., SCHARRER K., MURRAY T., BEVAN S. I., CHEN M.: Temporal visualization of boundary-based geo-information using radial projection. *Computer Graphics Forum 30*, 3 (2011), 981–990. 2

[DWL08] DODGE S., WEIBEL R., LAUTENSCHÜTZA K.: Towards a taxonomy of movement patterns. *Information visualization 7*, 3-4 (2008), 240–252. 2

[ENXS20] ELLIOTT M., NOTHELFER C., XIONG C., SZAFIR D.: A design space of vision science methods for visualization research, 2020. arXiv:2009.06855. 3

[Guo07] GUO D.: Visual analytics of spatial interaction patterns for pandemic decision support. *International Journal of Geographical Information Science 21*, 8 (2007), 859–877. doi:10.1080/13658810701349037. 2, 16

[JSM*16] JENNY B., STEPHEN D., MUEHLENHAUS I., MARSTON B., SHARMA R., ZHANG E., JENNY H.: Design principles for origin-destination flow maps. *Cartography and Geographic Information Science* (12 2016). 2

[KO09] KRAAK M.-J., ORMELING F. J.: *Cartography: Visualization of Spatial Data*, 3 ed. Routledge, 2009. 2

[LBR*16] LANDESBERGER T. V., BRODKORB F., ROSKOSCH P., ANDRIENKO N., ANDRIENKO G., KERREN A.: Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (January 2016), 11–20. doi:10.1109/TVCG.2015.2468111. 2, 16

[LCvL17] LUO D., CATS O., VAN LINT H.: Constructing transit origin–destination matrices with spatial clustering. *Transportation Research Record 2652*, 1 (2017), 39–49. URL: https://doi.org/10.3141/2652-05, doi:10.3141/2652-05. 2

[LJS20] L'YI S., JO J., SEO J.: Comparative layouts revisited: Design space, guidelines, and future directions, 2020. arXiv:2009.00192. 3

[LWLY15] LU M., WANG Z., LIANG J., YUAN X.: Od-wheel: Visual design to explore od patterns of a central region. *IEEE Pacific Visualization Symposium 2015* (07 2015), 87–91. 2

[Mac04] MACEACHREN A.: *How Maps Work: Representation, Visualization, and Design*. Guilford Press, 2004. 2

[NB13] NOCAJ A., BRANDES U.: Stub bundling and confluent spirals for geographic networks. In *Graph Drawing* (Cham, 2013), Wismath S., Wolff A., (Eds.), Springer International Publishing, pp. 388–399. 2

[NK16] NUSRAT S., KOBOUROV S.: The state of the art in cartograms. *Computer Graphics Forum 35*, 3 (2016), 619–642. 2

[PXY*05] PHAN D., XIAO L., YEH R., HANRAHAN P., WINOGRAD T.: Flow map layout. In *INFOVIS '05: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization* (2005), IEEE Computer Society, p. 29. 2

[Rob55] ROBINSON A. H.: The 1837 maps of henry drury harness. *Geographical Journal 121* (1955), 440–450. 2

[Rob67] ROBINSON A. H.: The thematic maps of charles joseph minard. *Imago Mundi 21*, 1 (1967), 95–108. 2, 16

[Sha48] SHANNON C. E.: A mathematical theory of communication. *Bell System Technical Journal 27* (1948), 379–423. 13

[SKD14] SLINGSBY A., KELLY M., DYKES J.: Featured Graphic. OD Maps for Showing Changes in Irish Female Migration between 1851 and 1911. *Environment and Planning A 46*, 12 (December 2014), 2795–2797. 2, 16

[SKDW12] SLINGSBY A., KELLY M., DYKES J., WOOD J.: Od maps for studying historical internal migration in ireland. In *IEEE Conference on Information Visualization (InfoVis)* (2012). URL: https://openaccess.city.ac.uk/id/eprint/1312/. 2

[SPR*18] SLINGSBY A., PATERSON A., RIGBY M., GRACE K., NGUYEN P., PERIN C., TURKAY C., ALJASEM D.: Characterising farms by the movement of animals through them. In *Proc. VIS 2018 Posters* (2018). 2

[Sta21] STATISTICS NETHERLANDS: More insight into mobility with the doughnut map. https://www.cbs.nl/en-gb/over-ons/innovation/project/more-insight-into-mobility-with-the-doughnut-map, 2021. 2, 9

[Tob87] TOBLER W.: Experiments in migration mapping by computer. *Cartography and Geographic Information Science - CARTOGR GEOGR INF SCI 14* (04 1987), 155–163. 2

[Tob04] TOBLER W.: Thirty five years of computer cartograms. *Annals of the Association of American Geographers 94*, 1 (2004), 58–73. 2

[VBS11] VERBEEK K., BUCHIN K., SPECKMANN B.: Flow map layout via spiral trees. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec. 2011), 2536–2544. URL: http://dx.doi.org/10.1109/TVCG.2011.202, doi:10.1109/TVCG.2011.202. 2, 16

[VCI20] VIOLA I., CHEN M., ISENBERG T.: Visual abstraction. In *Foundations of Data Visualization*, Chen M., Hauser H., Rheingans P., Scheuermann G., (Eds.). Springer, 2020. 15

[WDS10] WOOD J., DYKES J., SLINGSBY A.: Visualisation of origins, destinations and flows with od maps. *Cartographic Journal, The 47* (05 2010), 117–129. doi:10.1179/000870410X12658023467367. 2, 16

[YDGM17] YANG Y., DWYER T., GOODWIN S., MARRIOTT K.: Many-to-many geographically-embedded flow visualisation: An evaluation. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (Jan 2017), 411–420. doi:10.1109/TVCG.2016.2598885. 2, 17

[YWZ*19] YAO X., WU L., ZHU D., GAO Y., LIU Y.: Visualizing spatial interaction characteristics with direction-based pattern maps. *Journal of Visualization 22* (01 2019). doi:10.1007/s12650-018-00543-4. 2