DOI: 10.1111/cgf.70293

COMPUTER GRAPHICS *forum*
Volume 0 (2025), number 0, e70293

FluidMap: Proportional and Spatially Consistent Layout Enrichments in Multidimensional Projections

Daniela Blumberg, Patrick Paetzold, Michael Stroh, Oliver Deussen, Daniel A. Keim and Frederik L. Dennig

University of Konstanz, Konstanz, Germany
{daniela.blumberg, patrick.paetzold, michael.stroh, oliver.deussen}@uni-konstanz.de, keim@dbvis.inf.uni-konstanz.de, frederik.dennig@uni-konstanz.de

Abstract

Layout enrichment methods for multidimensional projections aim to enhance 2D scatterplots with additional information. We address the representation of categorical attributes with respect to a numerical feature, like the importance of a data point or its frequency, by colouring the scatterplots' background. However, applying existing space-filling methods has limitations: Voronoi partitionings, including weighted variants, do not correctly account for the relative weight of data points, resulting in disproportionately small or large areas, depending on the data point density. Neighbourhood Treemaps (Nmap) preserve the relative size of areas given data point weights but are restricted to rectangular shapes, often positioned far from the associated data points. To address these issues, we propose FluidMap*, a space-filling layout enrichment inspired by fluid dynamics. Our algorithm simulates the behaviour of coloured fluids spreading under pressure, with projected data points serving as sources and weights determining the amount of fluid to be distributed.* FluidMap *generates flexibly shaped areas that maintain sizes proportional to data point weights and include their assigned data points. We compare our method to Voronoi-based techniques and Nmap by quantifying their visual properties. Additionally, through an expert study, we assess task-specific differences. Our method outperforms existing techniques in preserving proportional representation and spatial consistency simultaneously.*

Keywords: visualisation, information visualisation

CCS Concepts: • Human-centred computing → Visualisation;

1. Introduction

Projection methods are well established for the visual analysis of high-dimensional data. They map the data into a lower-dimensional space, typically 2D, so it can be visualised, for example, by scatterplots. Projections aim to preserve the relationships between high-dimensional data points in a two-dimensional embedding [Jol86], enabling visual detection of meaningful structures like clusters. However, in addition to their position in the high-dimensional space, these data points often have associated numeric values, such as weights, and categorical attributes, such as class labels. The discrete nature of categorical attributes makes it hard to project mixed data [DJP*24, MGU*21]. To address this, layout enrichment methods visually augment the layout to improve the interpretability and insightfulness, for example, by representing the frequency of a categorical attribute in the background of the projection. Layout enrichments for 2D scatterplots comprise various techniques such as adding annotations like simple text [CLRP13], utilising colours to encode additional information [BTB13, LA11, Ves99], modifying the visual representation of data points based on specific attributes or metrics [DJP*24, CGSQ11], or incorporating interactive features like tooltips [dSRM*15, BTB13]. According to the categorisation scheme of Nonato and Aupetit [NA19], layout enrichments either focus on information from the original space or the projected space. Mapping information from the original space to points or neighbourhoods of the projected space is most commonly used. We address background enrichments in a two-dimensional layout to encode categorical attributes and their frequencies. Among others, background enrichment methods can be used to visually detect categorical attributes contributing to the neighbourhood formation [DJP*24].

To enrich scatterplot layouts with attributes of individual data points, a well-established approach is to apply Voronoi partitioning [Aur91] to the two-dimensional projection space, based on the locations of the data points [DJP*24, BWT*24, BTB13, LA11, Aup07]. Each Voronoi cell is determined by assigning the area to the closest data point and then coloured according to the attribute of the data point it is assigned to. To include data point weights,

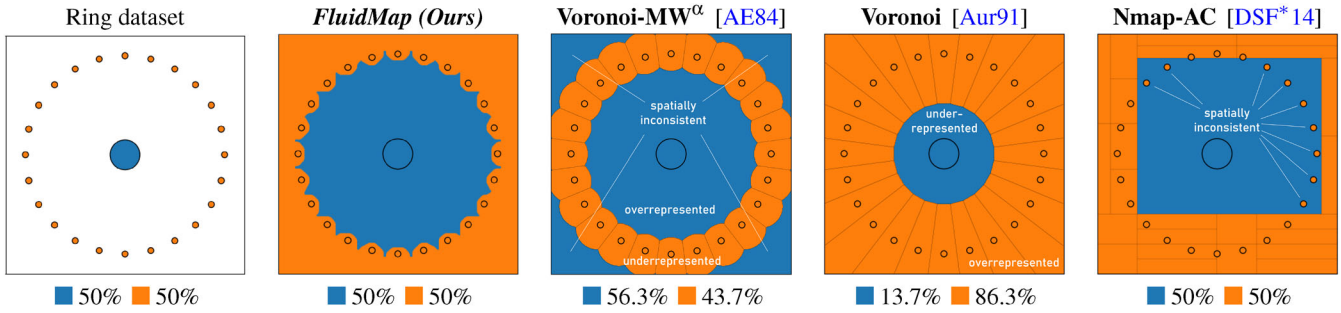


Figure 1: FluidMap *faithfully represents the frequency of an attribute and preserves spatial consistency. Current space-filling methods over- or under-represent attribute categories (i.e., all Voronoi-based methods), sacrifice spatial consistency (i.e., Nmap) or both (i.e., Voronoi-MW$^{\alpha}$).*

several adjustments to the Voronoi partitioning have been proposed [For86, Aur87, AE84]. However, regardless of the weight adjustment, Voronoi cells in sparse regions remain disproportionately larger than those in dense areas, resulting in undue visual emphasis. In contrast, Nmap [DSF*14], a treemap-based method, addresses over- and underrepresentation by mapping the data point weights directly to the size of rectangles. However, the resulting layout is difficult to interpret since Nmap places the rectangles independently of the actual point positions. Thus, the visual relationship between the data point and its area is lost. Finally, interpolation-based approaches are used to create layout enrichments [MBC*23, dSRM*15, MCMT14]. These methods assign space, usually pixels, in the background to multiple data points proportional to the distances. However, interpolation prevents estimation of relative area sizes, as the contributions of single data points are blended together.

Hence, we propose *FluidMap*, a space-filling background enrichment, which addresses the shortcomings mentioned above. *FluidMap* aims to preserve three main aspects: (1) The visual area in the layout assigned to a data point shall be proportional to the data point's weight, (2) each area shall extend around its assigned data point(s) such that it includes the point(s) itself and (3) the position of the projected points in 2D and the borders of the scatterplot shall not be altered. In this paper, we contribute:

- A novel layout enrichment technique for scatterplots that preserves *spatial consistency* of data points and their assigned areas as well as *proportional* relationships between area sizes and data point weights.
- A *qualitative* and *quantitative* evaluation comparing our method with existing space-filling approaches for background enrichment.
- A domain *expert study* to evaluate the practicality of our method for categorical data analysis.

2. Related Work

Layout enrichments relate to map-like visualisations, for which Hogräfer *et al.* [HHS20] provide an extensive overview and categorisation. Among these, layout enrichment techniques focusing primarily on a plot's background are termed *area imitating*. Methods of this category can be used to mitigate issues of scatterplots, for example, introduced by overplotting. Data point contours [MG13, JHM*13] or grid-based hexagonal bins [CLNL87] can be used in the background of the plot to express point densities. Alternatively, remapping of data points can be used to alleviate problems stemming from large differences in point density, such as dense areas frequently suffering from overplotting [RML24, RGE19]. We focus, however, on background enrichment without any remapping of data points, thereby preserving the original data distribution and avoiding distortions. Additional categorical attributes associated with the data points are visualised in the scatterplot background, similar to a cartographic map, where each region is assigned to an attribute.

We identified two classes of space-filling background enrichments:

Exclusive Mappings assign a certain space of the visualisation's background to exactly one data point in a one-to-one mapping. A prominent technique in this class is a *Voronoi* diagram [Aur91], that partitions a plane into regions, called *cells*, based on the closest proximity to a specified set of points in that plane, called *sites*. Each cell is bordered by a polygon of pixels that have an equal distance to at least two sites or belong to the boundary. Using the Voronoi partitioning for layout enrichment, one can colour the cells according to the attribute of the associated data points. In a Voronoi diagram, the size of a cell depends on the relative position of its associated site to the positions of neighboring sites. *Weighted* Voronoi diagrams can be used to influence the size of cells by incorporating the data points' weights into the distance function, for example, additively weighted (Voronoi-AW) [For86], power weighted (Voronoi-PW) [Aur87], or multiplicatively weighted Voronoi (Voronoi-MW) [AE84].

Voronoi diagrams have been widely applied to enhance the layout of projected multivariate data. As the projection of data into lower-dimensional space can introduce a variety of distortions, several Voronoi-based layout enrichments investigate the reliability of multidimensional projections by highlighting areas of distortions in the projection layout [BWT*24, HFA17, LA11, Aup07] and employ the Voronoi cell divisions as visual uncertainty measures [SvLB10]. Another application of Voronoi partitioning as background enrichment aims to visualise the influence of data points on the spatial distribution of categories and clusters within scatterplots. By colouring the cells according to a categorical attribute associated with the data points, Voronoi partitioning enables users to pre-attentively identify groups of similar data items and analyse the contribution of an attribute to neighbourhood formation in the projection [DJP*24,

BTB13]. This approach is even efficient for datasets with numerous categories where Voronoi cell groupings reveal shared local category combinations [DJP*24]. Interactive techniques that allow users to merge Voronoi cells have further improved the visual representation of categorical segmentation by enabling users to refine and enhance category assignments [HFA17, BTB13]. Despite their wide use as background enrichment, Voronoi partitioning cannot guarantee to represent weights in terms of cell sizes faithfully without changing the positions of the points in the scatterplot even when using a weight-adjusted distance function (see, e.g., Table 1). This typically leads to an overly high visual emphasis on data points associated with large cells in sparse regions, fundamentally violating the data-to-ink ratio [Tuf92].

Like Voronoi partitioning, treemaps assign visual areas to data points but they rely on recursive subdivisions. Johnson and Shneiderman [JS91, Shn92] first introduced the concept of treemaps using nested rectangles to visualise hierarchical structures. Since then, numerous space-filling techniques have been developed, each defining different strategies for recursive subdivision. Examples include the use of Voronoi cells [BDL05] and space-filling curves [Wat05]. An overview of treemap layout techniques is given in [SLD20]. *Neighborhood Treemap (Nmap)* [DSF*14] is a treemap algorithm that incorporates the neighbourhood of data points into the placement of assigned rectangles, making it a potential candidate for layout enrichment of scatterplots. This algorithm assigns a proportionally sized rectangle to each data point through horizontal or vertical splits while preserving neighbourhood relationships. Two variants based on the slice-and-scale process exist: Alternate Cut (Nmap-AC) alternates between horizontal and vertical splits, while Equal-Weight (Nmap-EW) improves the aspect ratio of the rectangles by using the direction of the split that produces more balanced bisections in terms of weights. Despite Nmap preserving the area assigned to each data point, it violates the Gestalt Principle of Common Region [Tod08] since rectangles are often placed apart from their assigned data point, leaving the data point contained by another rectangle. In contrast, our approach is designed to preserve spatial consistency, as each data point is included in its associated region, while still adhering to proportionality constraints.

Fuzzy Mappings assign a given space to multiple data points at once with varying impact in a one-to-many mapping. This class of techniques, often used as layout enrichment, is based on *interpolation* [MBC*23, dSRM*15, MCMT14]. They assign pixels in the background to multiple data points weighted inversely by distance. If a pixel is directly placed on a data point, it is assigned the attribute of this data point. Otherwise, each pixel is assigned the weighted sum of all attributes, with points that are closer having a greater influence. An exponent p is applied to the distance between a pixel and a data point to control the influence of a data point's attribute on the weighted sum. For $p \to \infty$, this converges to the standard Voronoi partitioning. A special case of interpolation techniques is Shepard interpolation, where only a certain radius around the data points is assigned, leaving many unassigned white spaces in the background, which makes it not space-filling.

In the context of evaluating multidimensional data projections, interpolation has been used as background enrichment to visually communicate distortions in the distances and neighbourhoods introduced by projection methods [MMT15, MCMT14]. The specific

Table 1: *Attribute-dependent quality measures for different layout enrichment methods computed on the evaluation datasets.*

	distance preservation				quantity preservation				neighborhood ($k = 3$)				spatial consistency (point | area)			
Dataset	FluidMap	Voronoi	Voronoi-MW$^\alpha$	Nmap-AC	FluidMap	Voronoi	Voronoi-MW$^\alpha$	Nmap-AC	FluidMap	Voronoi	Voronoi-MW$^\alpha$	Nmap-AC	FluidMap	Voronoi	Voronoi-MW$^\alpha$	Nmap-AC
Moons	.98	**1**	.99	.9	**1**	.99	.99	**1**	**1**	.96	.95	.97	**1** | **1**	**1** | **1**	**1** | **1**	.99 | .99
Unequal Pair	.67	**1**	.99	.67	**1**	.47	.47	**1**	**.98**	**.98**	.96	.96	**1** | **1**	**1** | **1**	**1** | **1**	.98 | **1**
Unequal Trio	.89	**1**	**1**	.9	**1**	.54	.56	**1**	**.97**	**.97**	**.97**	.94	**1** | **1**	**1** | **1**	**1** | **1**	.99 | .97
Equal Five	.99	**1**	**1**	.97	**1**	.93	.94	**1**	**1**	.96	.95	.93	**1** | **1**	**1** | **1**	**1** | **1**	.99 | **1**
Ring	.64	**1**	.57	.99	**1**	.43	.88	**1**	.68	.68	.68	**.92**	**1** | **1**	**1** | **1**	**1** | .52	.97 | **1**
Blobs	.73	**1**	**1**	.71	**1**	.29	.29	**1**	**.92**	.83	.83	.82	**1** | **1**	**1** | **1**	**1** | **1**	.89 | .89
Titanic	.74*	**1***	.85*	.77*	**1***	.54*	.71*	**1***	.63*	**.87***	.73*	.84*	**1*** | **1***	**1*** | **1***	**1*** | .84*	.9* | .94*
Property Sales	.82*	**1***	.89*	.86*	**1***	.58*	.81*	**1***	.74*	**.84***	.58*	.72*	**1*** | **1***	**1*** | **1***	**1*** | .94*	.93* | .94*

Note: Values are rounded to the second decimal place. For the FluidMap approach, a grid size of 100×100 is used without applying postprocessing, the pressure increase factor γ is set to 10%, $\alpha = 0.1$ for the Ring dataset, and $\alpha = 2$ for all others. For Voronoi-MW$^\alpha$, α is set to 2. Values marked with * indicate average values, where the individual values for each attribute are provided in the Supporting Information.

choice of projection method and its input parameters has a significant impact on embedding quality. This quality, as, for example, determined by a user study, can be visualised through colour interpolation [MBC*23]. Further, Shepard and nearest-neighbour (Voronoi) interpolation is used to investigate which dimension of the high-dimensional space is most important to describe point neighbourhoods [dSRM*15]. Yet, interpolation obscures the proportional aspects of the background enrichment as the contributions of single points are blended together. This makes it challenging to visually recognise the relative weights of the data points. In contrast, our proposed method aims to fill the whole background of the projection without interpolating colours or leaving white spaces, thus leveraging the entire projection space.

3. Method

FluidMap is designed to tackle the limitations of (weighted) Voronoi and Nmap by creating spatially consistent areas in the background of a scatterplot with area sizes representing data point weights.

Notation. Assume an n-dimensional dataset $D = \{x_i\}_{1 \le i \le |D|}$ with data points $x_i \in \mathbb{R}^n$. These data points have associated weights $W = \{w_i\}_{1 \le i \le |D|}$ with $w_i \in \mathbb{R}^+$. Weights can represent frequencies, such as duplicate data points prominent in categorical data analysis, or indicate the importance of a data point. If no weights are explicitly given, then usually each point has the same weight, that is, $w_1 = \cdots = w_{|D|}$. Further, let $A = \{a_i\}_{1 \le i \le |D|}$ denote the data points associated attribute values to be encoded in the background. A projection technique P maps D to a lower dimensional space $D' := P(D) = \{x'_i = P(x_i) | x_i \in D\}$. Here, we assume $D' \subset \Omega \subset \mathbb{R}^2$ such that the transformed data can be plotted in 2D, with Ω denoting the rectangular area of the scatterplot, commonly a square.

Problem Statement: Given the set $P(D)$ of 2D points x'_i with an associated attribute a_i and a weight w_i and the set $\Omega \subset \mathbb{R}^2$ of pixels q representing the background of the 2D scatterplot, we aim to assign each pixel to exactly one data point and colour it according to the point's attribute while satisfying the following constraints:

- **C1** The size of the area assigned to a data point should be relative to its weight.
- **C2** The extent of an area should remain close to its assigned data point(s) and include the data point(s) itself.
- **C3** The position of the data points in 2D and the borders of the scatterplot (given by Ω) are fixed and should not change.

To evaluate our proposed method, we develop quality metrics based on these constraints, as most of them can be made measurable. Methods for creating layout enrichments using this definition exist (see Section 2). However, these approaches do not cover all the constraints **C1**–**C3** simultaneously.

We base our algorithm on the metaphor of fluids spreading towards the lowest external pressure, allowing fluids with the same attribute value to merge. We thereby consider the projected data points as the sources of fluid dispersion and the data point weights as indicators of the quantity of each fluid to be distributed. Similar to their metaphorical counterparts, we assume fluids of higher quantity to spread faster than fluids of lower quantity. Further, fluids are usually most concentrated at the point of origin, and as they disperse, their intensity diminishes the further they spread out. While spreading, a fluid may encounter resistance from the environment, causing it to spread more in directions of lower resistance. To control the fluid spread, we introduce the concept of *pressure*, which models the influence of a data point on surrounding pixels.

3.1. Measuring Pressure

We assume that the projected data points x'_i exert different levels of pressure on the pixels in Ω, determined by two factors: (1) The pressure depends inversely on the Euclidean distance d of the projected point to the respective pixel. This reflects the natural gradient of the fluid's concentration as it spreads further away from its source. (2) The pressure is influenced by the associated weight of the data point, with higher weights exerting more pressure, thereby considering the larger amount of fluid originating from the point to be distributed. It is important to emphasise that the term *pressure* here is not intended as a direct analogue to physical pressure but serves as an analogy. Inspired by the idea of Voronoi-MW [AE84], we define the relationship between distance and weight as a ratio of these two quantities. In addition, we introduce the parameter α to control the relative influence of distance and weight. Formally, the pressure of point x'_i on a pixel q is defined as

$$\mathit{pressure}\left(x'_i, q\right) = \frac{w_i}{d\left(x'_i, q\right)^{\alpha}}, \tag{1}$$

where the exponent α can regulate how sharply pressure decays with distance. A value of $\alpha = 1$ can be seen as the neutral case, with larger values of $\alpha > 1$ increasing the penalty for distance, leading to more locally concentrated regions, while smaller values ($0 < \alpha < 1$) allow distant points to exert broader influence. Thus, α determines whether proximity or weight plays the dominant role in shaping the regions. An exponent is applied asymmetrically only to the distance and not to the weight, due to their different value ranges. In our setting, distances are mostly greater than 1 for non-source pixels, while weights are typically normalised between 0 and 1 to sum up to 1. Exponentiation affects these ranges very differently. Also, while weight varies only with respect to data points, distances depend on points and pixels, providing a finer-grained tuning. Hence, we apply α to the distance only. We can visualise the mean overall pressure by taking the average exerted pressure for each single pixel (Equation 2) and colour Ω accordingly (see supplementary material). This provides a first intuition on the established fluid landscape, the prevailing pressure ratios, and the influence of α. To further examine the role of α in shaping the layout enrichment, we conducted a parameter sensitivity analysis (see Section 3.3).

$$\overline{\mathit{pressure}}(q) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{w_i}{d\left(x'_i, q\right)^{\alpha}}. \tag{2}$$

3.2. Simulating Fluids

The input to the *FluidMap* algorithm is the set of projected data points $P(D)$ sorted ascendingly by weight, their respective attributes A and weights W, the set of pixels Ω, the pressure tuning parameter α, and a pressure increase factor $\gamma > 0\%$. The projected data and the pixel space are normalised such that the width and height of a single

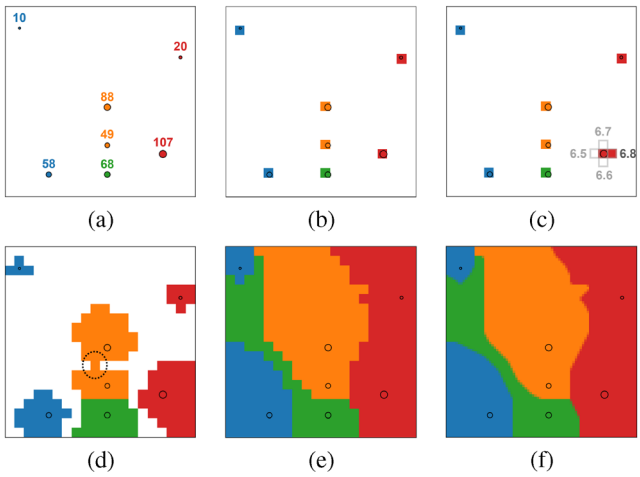


Figure 2: *Overview of the* FluidMap *algorithm on a simple example dataset with a* 20×20 *pixel grid (a). Colour encodes attributes, point sizes and numbers on top represent weights. (b) Initially, we assign a fixed source pixel to each data point based on the minimum point-pixel distance. (c) During each iteration, we assign the candidate pixel with the highest pressure ratio (gray numbers) to a given target point (determined by the highest pixel deficit). (d) When areas of the same attribute connect, we merge the groups of the corresponding data points. (e) The assignment process is repeated until ideally all pixels are assigned to data points. (f) Post-processing can be applied to reduce pixelation, resulting in a smoother appearance.*

pixel equal a unit of 1. The algorithm iteratively assigns each pixel $q \in \Omega$ to one data point as follows and as shown in Figure 2. Pseudocode of *FluidMap* can be found in the Supporting Information.

Step I: Initialisation

We assign the nearest unassigned pixel to each data point (i.e., the smallest Euclidean distance of the data point and pixel centre) as its *source* pixel and block this pixel completely from any reassignment (Figure 2b). This is crucial to ensure that each fluid spread has a fixed origin. As the input data points are sorted by their weights, we favour low-weight points by selecting their source pixel first. However, as the size of Ω increases, the order of source pixel assignments makes no noticeable difference anymore, as conflicts between close data points vanish. As we allow regions with the same attribute to merge, we track the merging and splitting of regions using group labels. Initially, each data point starts as its own group, and once some areas of the same attribute get connected, these points are given the same group label (Figure 2d).

Step II: Iterative Assignments using Pressure Ratio

We then repeat the following steps until, ideally, every pixel is assigned to a data point. In each round, we calculate the pixel deficit of each data point. Denoting the expected number of assigned pixels given by the weight of the data point as $\tilde{m}_i = \frac{w_i}{\sum_{j=1}^{|W|} w_j} |\Omega|$ and the current number of assigned pixels as m_i, we define the pixel deficit as $\frac{\tilde{m}_i - m_i}{\tilde{m}_i}$. We then select the data point with the largest pixel deficit to simulate that fluids of higher quantity spread faster than fluids of lower quantity. All pixels adjacent to the boundary of the area assigned to the group of the selected target point are considered candidates for assignment. To decide which of these candidates to assign to the target point x_t', we calculate the pressure ratio for each candidate q and select the pixel with the highest positive pressure ratio (Figure 2c). This pressure ratio is defined as

$$pressureRatio\left(x_t', x_c', q\right) = \frac{pressure\left(x_t', q\right) - pressure\left(x_c', q\right)}{\overline{pressure}(q)}, \tag{3}$$

where $pressure(x_t', q)$ denotes the pressure (as defined in Equation 1) of the target point x_t' on the candidate pixel q, that is, target pressure, and $pressure(x_c', q)$ the pressure of the data point the pixel is currently assigned to, that is, current pressure. If the pixel is currently unassigned, the current pressure equals 0. We divide the difference of pressures by the average pressure exerted by all data points on the candidate pixel (Equation 2) to push the spreading of areas towards regions of low external pressure, corresponding to, for example, low-density regions.

If we find a valid target pixel, that is, a pixel adjacent to the assigned area of the target group for which the pressure ratio is positive, we (re)assign it to the target data point and update the group labels if areas with the same attribute merge. If the target pixel was previously assigned to another point, we block that pixel from being reassigned to the previous point, reducing back-and-forth swapping of assignments. Swapping assignments can cause areas to become disconnected from their sources. As soon as we detect a disconnection, we update the group labels and unassign any pixels that are disconnected from their assigned data point. In doing so, we ensure that the number of areas associated with a certain attribute does not exceed the number of data points sharing that attribute, so that no area becomes spatially disconnected from its source. If no valid target pixel can be found during an iteration, this indicates that the pressure of the target point's group is too weak, resulting in a pressure ratio ≤ 0. In that case, we increase the pressure of all data points in the respective group by the factor γ and unblock blocked non-source pixels. Updating the pressure values ensures that each data point receives an area proportional to its weight. Specifically, pressure is shifted toward groups of data points that cannot spread out easily, for example, when the space around them is limited. We can inspect how the pressure shifts by visualising the mean pressure.

3.3. Parameter Choice and Sensitivity

FluidMap has three parameters: the pressure tuning parameter α, the size of the pixel grid $|\Omega|$ and the pressure increase factor γ. We conducted an in-depth analysis of the sensitivity of *FluidMap*'s generated layouts to these parameters, both visually and in terms of runtime and quality measures (see Section 4), in order to provide recommendations for their values. To determine the influence of individual parameters, we kept all others constant. The detailed results in the Supporting Information show that α has the highest visual influence, with low values tending to produce more spilled-out areas as the distance from the data points increases, similar to fluids spreading with low cohesion. In addition to visual assessment of the final layouts, mean pressure visualisations can be used before executing the algorithm to determine a reasonable choice of α. For the evaluation datasets (see Section 4), we recommend using a

value of $\alpha = 2$, which strikes a reasonable balance between visual appearance, performance and quality. Ring is the only exception, where we suggest $\alpha = 0.1$. The choice of the grid size $|\Omega|$ presents a trade-off between the smoothness of generated shapes and runtime. While layouts with a lower resolution appear more pixelated, we found that patterns in the layout tend to remain consistent over different grid sizes. Thus, we suggest using a relatively low resolution (100×100) that exceeds the number of projected data points, for example, for our datasets by a factor of 147 on average (min. 16.7, max. 417), and applying postprocessing as described next in Section 3.4 for a smoother visual appearance. The pressure increase factor showed the least fluctuations. We use $\gamma = 10\%$ in our evaluation, with other values producing similar results. As *FluidMap* is explicitly designed to preserve quantity and spatial consistency, the algorithm is, in general, not sensitive towards any parameter changes with respect to these metrics. Thus, *FluidMap* produces consistently high quantity and spatial consistency scores. The other quality metrics and the visuals tend to remain stable in our analysis as well. Yet, the runtime can vary significantly between different parameter settings.

3.4. Postprocessing

Our sensitivity analysis revealed a trade-off between a short runtime and a high resolution of the pixel grid Ω. Thus, we suggest using a relatively low resolution (100×100) and applying the hq4x [Ste03] image upscaling algorithm as a postprocessing step (Figure 2f). We chose this method because it is specifically designed for high-contrast, pixelated images and does not require a memory-intensive neural network-based upscaling. In contrast, it works by detecting differences between pixels in the colourspace and expands a single pixel into a 4×4 block of pixels. Respective figures of the *FluidMap* layout enrichment without image upscaling for all evaluation datasets are provided in the Supporting Information, demonstrating that the overall visual patterns generated by *FluidMap* are not affected by this postprocessing step.

4. Results and Evaluation

To evaluate the effectiveness of our approach, we compare it with existing space-filling methods that use exclusive area assignments (i.e., one-to-one mappings). We selected classical Voronoi partitioning, weighted Voronoi partitioning and Nmap. Given a set of sites, here D', and a bounded, convex area, here Ω, the Voronoi cell $V(x'_i)$, associated with each data point is defined as the set of all pixels $q \in \Omega$ that are closer to x'_i than to any other point in D': $V(x'_i) = \{q \in \Omega : d(q, x'_i) < d(q, x'_j) \quad \forall x'_j \in D' \setminus x'_i, i \neq j\}$. In Voronoi-MW, the Euclidean distance d is substituted by the weight-dependent distance $\frac{\|q-x'_i\|}{w_i}$. As noted in Section 3.1, the objective function of Voronoi-MW is similar to our definition of pressure in *FluidMap*, but it does not include an α parameter to balance the relative influence of distance and weight. Thus, to enable a fair comparison, we propose to extend Voronoi-MW with such a parameter, so the distance function becomes $\frac{\|q-x'_i\|^\alpha}{w_i}$. Since this method, denoted as Voronoi-MW$^\alpha$, achieved the best performance among the weighted Voronoi variants with $\alpha = 2$, we report only the results for this weight adjustment. For Nmap, both versions perform on a comparable scale, so we only display the alternate cut version (Nmap-AC), which produces slightly better quality metrics than the equal-weight version (Nmap-EW). Respective plots and metrics for the remaining layout enrichments can be found in the supplementary material.

As evaluation datasets, we selected synthetically generated datasets as a baseline, including four datasets that are used in the evaluation of Nmap, and two real-world datasets to demonstrate an application case of *FluidMap*. The chosen configurations of Nmap each consist of 600 data points in 2D, whose weights were drawn from a Gaussian distribution. The number of distinct groups, as well as their sizes and densities, vary across the different configurations. (1) *Moons*, that is, configuration 2 in Nmap, comprises two interleaving half-moon-shaped clusters of equal size and density. (2) *Unequal Pair*, that is, configuration 5 in Nmap, resembles two clusters with different densities and sizes (500 vs. 100 data points). (3) *Unequal Trio*, that is, configuration 8 in Nmap consists of three adjacent clusters where the middle one is smaller and has a lower density than its neighbours (30 vs. 285 data points). It demonstrated how the methods behave with more than two groups. (4) Finally, *Equal Five*, that is, configuration 9 in Nmap, shows five approximately equidistant, same-sized and equally distributed clusters. Additionally, we generated two more synthetic datasets to evaluate the behaviour for other weight distributions: (5) *Ring* contains equally weighted data points arranged in a ring and one data point in the centre of the ring of a different class with a high weight difference to the ring points. The weights are chosen such that the centre point's weight corresponds to the sum of the weights of the other points. (6) *Blobs* contains 1000 data points drawn from a Gaussian distribution with 5 clusters and 100 additional randomly distributed noise samples projected from $n = 10$ to 2D using Multidimensional Scaling (MDS) [Kru64] and exemplifies a scenario where weights across all data points are exactly the same. As real-world datasets, we include two categorical datasets: *Titanic* [Daw95] provides information about passengers on the Titanic ship, including their survival status, gender, age and ticket class. *Property Sales* [KSDK11] comprises data on property sales in Singapore, including information about the current housing situation of purchasers and the type and location of the purchased property. Both categorical datasets include binary as well as multi-valued attributes. For all datasets under consideration, weights are normalised to represent ratios of total weight, summing up to 1.

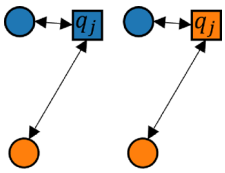
4.1. Quantitative Comparison

For a quantitative evaluation, we consider quality metrics that originate from related work [Aur91, DSF*14] or can be directly derived from the imposed layout constraints **C1–C3** (see Section 3). Each proposed metric can be defined on an *attribute-independent* level, where the direct assignment of pixels to data points is considered independently of the attribute of the data point, or on an *attribute-dependent* level, where the actually assigned data points are neglected by focusing on their attributes. The latter is less restrictive but reasonable from a visual perspective, since the actual data point to which a pixel is assigned is not visible in the layout, and only its attribute determines the pixel's colour. To align metric values to the visuals, we define and report *attribute-dependent*

metrics in the following. The respective counterparts can be found in the supplementary material.

Distance Preservation: This metric results directly from the objective of a standard Voronoi partitioning and measures if each pixel is assigned to its closest data point, or more generally, if a pixel is assigned the same attribute as its closest data point as

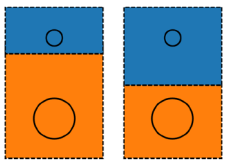
$$1 - \frac{1}{|\Omega|} \sum_{j=1}^{|\Omega|} \frac{\min_{x' \in D'_{A^{(j)}}} d(q_j, x') - \min_{x' \in D'} d(q_j, x')}{\max_{x' \in D'} d(q_j, x') - \min_{x' \in D'} d(q_j, x')} \in [0, 1], \quad (4)$$



Distance Preserving *Not Distance Preserving*

where $D'_{A^{(j)}} := \{x' \in D' \mid A(x') = A(x'^{(j)})\}$ and $A(x'^{(j)})$ denotes the attribute of the data point x' to which the pixel q_j is assigned. A value of 1 indicates that each pixel $q_j \in \Omega$ is assigned the same attribute as the projected data point to which it has the lowest Euclidean distance d, and a value of 0 indicates that each pixel is assigned the attribute of its farthest data point. Distance preservation is also an indicator of how well condition **C2** is fulfilled by the layout enrichment. As Nmap is not defined on a pixel basis, we adjust the metric to take the sum over rectangles instead of pixels.

Quantity Preservation: This metric measures the degree to which the size of the assigned areas reflects the weights of the data points (**C1**). From a more general visual perspective, it measures the degree to which the colours defined by the attributes of the assigned data points reflect the overall attribute distribution of the dataset.



Quantity Preserving *Not Quantity Preserving*

Let $\mathcal{A}$ denote the set of unique attribute values in A, $\pi^a_D = \frac{\sum_{i|a_i=a} w_i}{\sum_{i=1}^{|D|} w_i}$ the relative proportion of an attribute a in the dataset with respect to data point weights, and let $\pi^a_\Omega = \frac{|\Omega^{(a)}|}{|\Omega|}$ denote the relative size of the area of pixels assigned and coloured according to a in the visual layout. Then, attribute-dependent quantity preservation is formally defined as

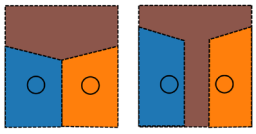
$$1 - \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \frac{\left|\pi^a_D - \pi^a_\Omega\right|}{\max\left(\pi^a_D, \pi^a_\Omega\right)} \in [0, 1]. \quad (5)$$

Dividing the absolute difference of expected and actual proportion by their maximum normalises the values in the sum to a range between 0 and 1. A metric value of 1 indicates that the number of pixels assigned to data points of the same attribute faithfully represents the sum of their weights.

Neighbourhood Preservation: We quantify to what degree neighbours of data points are preserved in the layout enrichment by observing the neighbourhood of the assigned areas. The metric is comparable to the k-nearest neighbour index used in the evaluation of Nmap and calculates the average percentage of k-nearest neighbours of data points that are preserved in the layout. To align this visually with the attributes again, we check if the attributes of the data point's neighbours are the same as the attributes of their neighbouring areas as

$$\frac{1}{|D|} \sum_{i=1}^{|D|} \frac{f_A\left(\mathcal{N}_k\left(x'_i\right)\right) \equiv f_A\left(\mathcal{N}_k\left(\Omega^{(i)}\right)\right)}{k} \in [0, 1], \quad (6)$$

where f_A returns the counts of the unique attributes of the given neighbours with $\mathcal{N}_k(x'_i)$ denoting the set of k-nearest neighbours of the projected data point x'_i and $\mathcal{N}_k(\Omega^{(i)})$ the set of k-nearest neighbours of its assigned area $\Omega^{(i)}$ in the spatial layout. We then compare their attribute counts, providing us with the number of shared attribute values. A value of 1 reflects a perfect preservation of neighbourhood relationships for a given value of k, indicating that the layout incorporates similarity relationships. We define neighbours in the layout as the data points associated with the neighbouring areas, that is, adjacent areas that share a border with the target area.



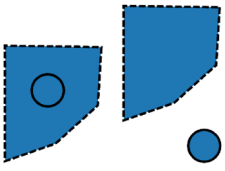
Neighborhood Preserving *Not Neighborhood Preserving*

A longer shared border thereby indicates a closer neighbourhood. Alternatively, one could compute the centroid of each area and determine neighbours based on the distance of the centroids, as done in Nmap. However, since area shapes vary strongly between the different layout enrichments under consideration and might be non-convex, we compute shared border lengths instead of centroid distances, addressing [MT20]. This also influences the range of meaningful k-values, as the number of directly adjacent areas that share a border with a target area is limited, as opposed to neighbours defined by distances. Thus, we decide to use a value of $k = 3$ in our evaluation.

Spatial Consistency: The following two measures evaluate how well the projected data points x'_i are spatially aligned with the assigned areas on the visualization plane Ω and vice versa (**C2**). Specifically, for the data points, we check whether each point is enclosed by its assigned area or, more generally, by an area that shares its attribute. This is given when at least one pixel assigned the same attribute as the data point has zero distance to that point:

$$1 - \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{\min_{q \in \Omega^{(a_i)}} d\left(q, x'_i\right)}{\frac{1}{2}\sqrt{2|\Omega|}} \in [-1, 1], \quad (7)$$

where $\Omega^{(a_i)}$ denotes the set of pixels q assigned the same attribute as the projected point x'_i and $d(q, x'_i)$ is their Euclidean distance.



Spatially Consistent *Not Spatially Consistent*

If the set of same attribute pixels is empty, the numerator is set to the denominator, so the fraction becomes 1. We divide the minimum distance of a point to a same coloured area by $\frac{1}{2}\sqrt{2|\Omega|} = \frac{1}{2}\sqrt{2}d(q_1, q_{|\Omega|})$ because we assume that the visualisation plane Ω is a square such that the length of its diagonal corresponds to $\sqrt{2|\Omega|}$, that is, the maximum possible distance in the plot. Taking half of the diagonal gives the metric double the value range, that is, $[-1, 1]$. Furthermore, we subtract the average from 1, such that a metric value of 1 indicates spatial consistency of data points, indicating each point is enclosed by

an area of its corresponding attribute. A negative value is theoretically possible but unlikely, as it would mean that all areas would be positioned on the opposite side of the plot than the data points of their assigned attribute, demonstrating a complete lack of spatial consistency, such as in a mirroring scenario where data points and areas are flipped relative to each other. The spatial consistency measure of data points is comparable to a special case of the displacement measure used in the evaluation of Nmap, which measures the average distance of a data point to the centroid of its assigned area. However, this more generally defined measure does not explicitly indicate if a point is enclosed by a respective area. Only in a scenario with zero displacements would the spatial consistency of points be 1 automatically. Additionally, our metric replaces the centroid distance with the minimum distance of a point to its area since evaluating distances to centroids does not make sense for non-convex areas [MT20].

Similarly to the spatial consistency of data points, we can also compute the spatial consistency of areas by checking whether each spatially disjoint continuous area of pixels that the colouring according to the assigned attribute produces includes a respective data point. This is given when at least one data point of the same attribute has zero distance to that area. Denoting $\mathcal{R}$ as the set of all spatially disjoint attribute regions that the layout produces, we can calculate the spatial consistency of areas as

$$1 - \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \frac{\min_{q \in r} \min_{x'_i \in D' | a_i = a_r} d(q, x'_i)}{\frac{1}{2}\sqrt{2|\Omega|}} \in [-1, 1]. \quad (8)$$

Table 1 displays these quality metrics calculated for different layout enrichment techniques and datasets. As expected, standard Voronoi has the highest distance preservation score, with a value of 1 for all datasets, since it directly incorporates the Euclidean distance as the objective function. The next best performance in terms of distance preservation is achieved by Voronoi-MW$^\alpha$. *FluidMap* and Nmap are in a similar range, although the values for Nmap are computed as the sum over rectangles instead of pixels. This computation tends to be in favor of Nmap, especially for datasets that produce rectangles with large areas. The quantity constraint is by design perfectly preserved in *FluidMap* and Nmap. For datasets where weight differences are evident across data points, Voronoi-MW$^\alpha$ scores a higher quantity preservation than the non-weight-adjusted standard Voronoi but still lower than *FluidMap* and Nmap. For the neighbourhood preservation with $k = 3$, no method clearly outperforms the others. However, standard Voronoi and *FluidMap* tend to achieve the highest scores. By design of the methods, *FluidMap*, as well as standard Voronoi, are spatially consistent, reflected by a measure of 1 for points and areas. While Voronoi-MW$^\alpha$ is also spatially consistent with respect to data points, it's generated areas in the layout are often spatially disconnected from their data points. For Nmap, spatial consistency in general tends not to be fulfilled. This indicates a clear trade-off between the metrics. Nmap is primarily optimised for preserving quantity, so it scores poorly on most other metrics. On the other extreme, standard Voronoi partitioning has perfect distance preservation, incidentally high neighbourhood preservation, and spatial consistency by design, but does not account for quantity preservation. While Voronoi-MW$^\alpha$ balances between quantity and distance preservation, it fails to achieve a perfect score for either and loses spatial consistency with respect to areas. *FluidMap* provides a powerful alternative that simultaneously guarantees quantity preservation and spatial consistency by design.

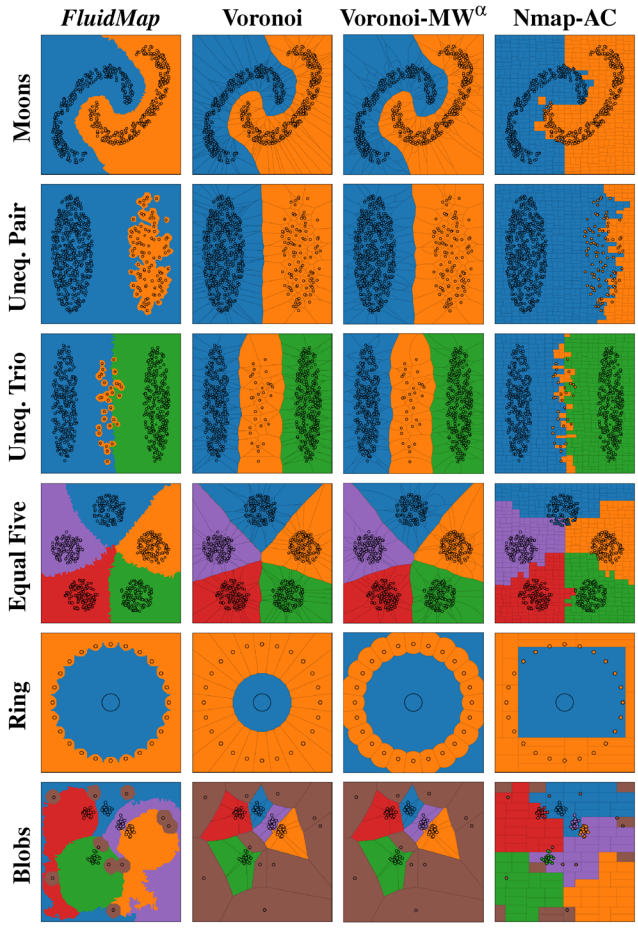


Figure 3: *Layout enrichments for six synthetic datasets. For FluidMap, a grid size of* 100×100 *is used with* $\gamma = 10\%$, $\alpha = 0.1$ *for the Ring data, and* $\alpha = 2$ *for all others. For Voronoi-*MW$^\alpha$, $\alpha = 2$.

4.2. Qualitative Comparison

In the previous section, we showed that *FluidMap* generates space-filling layout enrichments of high quality, regarding established measures. Thus, we continue to compare the visual appearance of our technique with the results of existing methods to provide a more detailed description of the results than general measures can. Figure 3 shows the different results when applied to the baseline datasets. Standard Voronoi and Voronoi-MW$^\alpha$ produce visually similar background colour segmentations for most datasets, except Ring. The influence of data point weights becomes visible only when examining the individual Voronoi cells closely. Voronoi-MW$^\alpha$'s borders tend to be curved, and the area sizes of individual data points differ more due to the weight adjustment. When no weight differences exist in the dataset (Blobs), a weight adjustment to the Voronoi objective function has no effect, collapsing to the standard version. Thus, attribute distributions can not be correctly inferred from the layout if the density of data points

differs across (groups of) data points. For datasets where weight differences are more pronounced (Ring), the final background encoding severely differs between the standard and weight-adjusted Voronoi: For Voronoi-MW$^{\alpha}$, the attribute of the point(s) with the highest sum of weights (coloured blue) receives more space in contrast to the standard Voronoi layout, where the other class (coloured orange) is more dominant. While the Voronoi-MW$^{\alpha}$ enrichment achieves a closer approximation to the true attribute distribution, it does not guarantee perfect quantity preservation since the assignment is still highly influenced by data point density. Furthermore, Voronoi-MW$^{\alpha}$ can generate spatially disconnected areas, that is, areas that contain no data points (Ring). When the subset size, summed weights and density among the clusters are approximately the same, Voronoi layout enrichments can achieve an overall attribute category segmentation that resembles the true attribute distribution (Moons and Equal Five) but only under exactly these circumstances. In contrast, *FluidMap* and Nmap both visually preserve the relationship between data point weights and area sizes, even with unevenly distributed attribute categories (Unequal Pair, Unequal Trio, and Blobs) or large weight differences across data points (Ring). However, when placing the areas, Nmap sacrifices spatial consistency completely in order to achieve proportional relationships of area sizes and weights. As a result, projected data points are not always enclosed by their own attribute region and vice versa, areas do not always include their assigned data points. This destroys the visual linkage between data points and assigned areas, becoming visually evident, especially when data points are positioned in differently coloured areas. *FluidMap*, in contrast, ensures that each data point remains within its own attribute area while still adhering strictly to proportionality. However, when background space is limited in highly populated regions, assigned areas might spread toward far-distant spaces (Blobs).

Another clear difference in visual appearances originates from the diversity in the generated area shapes. *FluidMap* imposes no restrictions on the shape apart from the requirement that the areas must remain connected to their assigned data point(s). Yet, this might enable jagged shape boundaries due to the simulated organic behaviour of fluids. In contrast, boundaries appear more straight for Voronoi-based and Nmap approaches, due to the geometric division of the plane. The restriction of Nmap to rectangular shapes becomes particularly prominent when the number of data points is low. While a large number of rectangles can approximate, for instance, a circular shape in highly populated layouts, a low number of data points for an attribute category leads to unflexible shapes restricted to rectangles (e.g., Ring). Further, Nmap struggles to preserve the symmetry of the scatterplot points (Equal Five and Ring). For all layouts, except Voronoi-MW$^{\alpha}$, the number of spatially disjoint areas for an attribute does not exceed the number of data points sharing the attribute. For Voronoi-MW$^{\alpha}$, this property is not guaranteed, since one point's area can split into multiple disconnected areas (Ring). The visual results also demonstrate that *FluidMap* is better at handling sparsely populated regions containing outliers (Blobs).

In summary, the visual assessment, combined with the provided quality metrics, demonstrates that *FluidMap* generates quantity preserving and spatially consistent layout enrichments, which are adaptable to various scatterplot layouts due to the flexibility in area shape. Unlike Nmap, which sacrifices spatial consistency to achieve proportionality, or Voronoi-based methods, which do not guarantee proportionality between area sizes and data point weights and might produce spatial inconsistency with respect to areas, *FluidMap* ensures that data points and corresponding attribute areas are spatially consistent while preserving proportional relationships.

4.3. Expert Study: Categorical Data Projections

Each layout enrichment method optimises for different aspects of the underlying data. So far, we have compared these methods in terms of quality measures and described the visual patterns they generate for baseline datasets. However, their effectiveness and usability highly depend on the application and goal of the visualisation. Thus, in the following, we present a use case of *FluidMap* for the analysis of two real-world categorical datasets. Layout enrichment can be used for the similarity-based analysis of categorical data by projecting it first and then segmenting the visual plane according to a given user-selected categorical attribute [DJP*24]. This enables the identification of similar subsets in the data and the detection of attributes that strongly influence the topology of the 2D projection. We conducted an expert user study in German with five data analysts, E1–E5, all male, with an age range between 25 and 31. All have a strong background in information visualisations and are currently pursuing a Ph.D. The experts have high experience at least in either categorical data analysis or multidimensional projections. The study started with an introduction to background enrichment, demonstrated on the *Ring* dataset (see Figure 1). This was followed by instructed tasks to be performed on projections of the *Titanic* and *Property Sales* datasets, visually enriched by FluidMap, Voronoi, Voronoi-MW$^{\alpha}$ and Nmap-AC (Figures 4 and 5). We apply the same projection as Dennig *et al.* in [DJP*24] to enable a direct comparison. To resemble realistic analysis tasks, we used same-sized black dots to represent data points, that is, colour and size independent of their frequency and attribute categories. Doing so, conclusions about the dataset characteristics must be drawn from the background colouring. For a given task, one layout enrichment method was presented at once, where its name was replaced with a letter. For each task and participant, we randomised the order of the methods. After solving the given task for all methods individually, the expert was presented with the ground truth (if present in the data). He was then asked to reflect on how well he performed with each method in solving the task. We used four tasks concerning individual and groups of data points, attribute distributions, and the topological structure:

- **T1** Identifying the attributes of a single data point.
- **T2** Identifying common attributes of a group of data points.
- **T3** Estimating the distribution of attribute categories.
- **T4** Determining if and how attributes influence the topology of the projection.

Throughout the study, participants were asked to verbalise their thoughts and to comment on their preferences regarding the four presented methods. An overview of the study setup is provided in the Supporting Information. Ethical approval is not required, since the study is evaluating the effectiveness of FluidMap.

T1 & T2: We asked the experts to identify the attributes of a single highlighted data point and a group of highlighted data points. All methods except Nmap supported this successfully. Nmap's placing

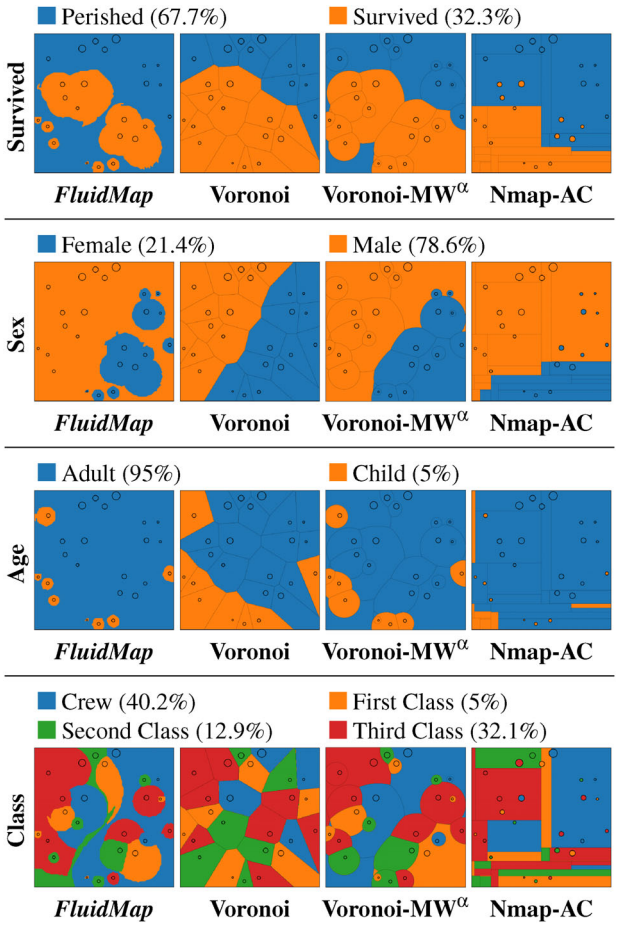


Figure 4: *Layout enrichments for different attributes of the Titanic data [Daw95]. For FluidMap, a grid size of* 100×100 *is used with* $\gamma = 10\%$ *and* $\alpha = 2$. *For Voronoi-*MW$^{\alpha}$, $\alpha = 2$.

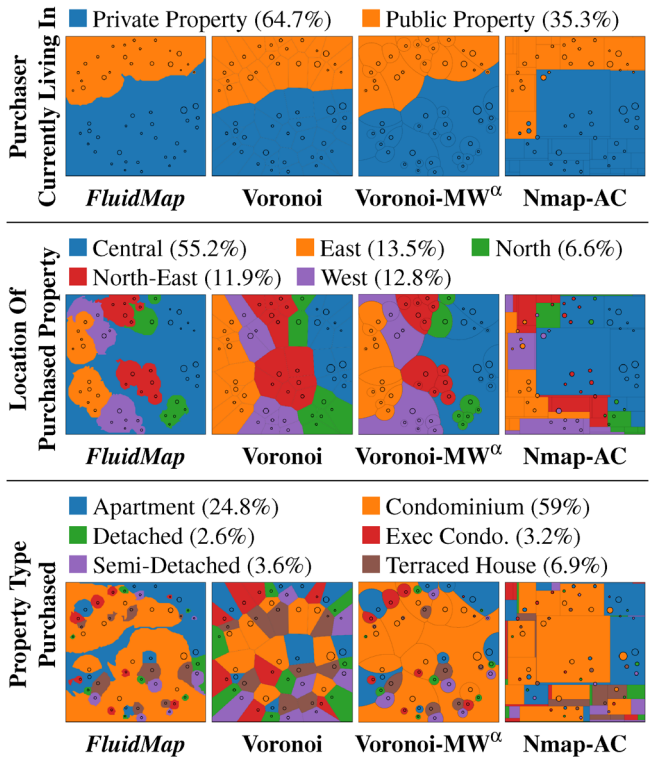


Figure 5: *Layout enrichments for different attributes of the Property Sales data [KSDK11]. For FluidMap, a grid size of* 100×100 *is used with* $\gamma = 10\%$ *and* $\alpha = 2$. *For Voronoi-*MW$^{\alpha}$, $\alpha = 2$.

of rectangles independently of the exact data point positions violates spatial consistency, losing the visual relationship of which area corresponds to which data point(s). Experts found this confusing and lost trust in the method. All other methods performed similarly well with some nuances. E3 mentioned that the Voronoi enrichment made it difficult to identify target points or groups. This is in line with the observation of E4, who found it difficult to mentally separate clusters for multi-valued attributes (e.g., `Class` or `Property Type Purchased`). Yet, he found the cell structure helpful to classify single data points. E1 positively highlighted the presence of similar shapes around groups in *FluidMap* compared to all other methods, simplifying cluster identification across attributes. The enrichments of the categorical data projections also demonstrate the similarities between *FluidMap* and Voronoi-MW$^{\alpha}$ in facilitating circular shapes. While E5 preferred Voronoi's larger areas for classifying clusters, he appreciated island-like regions in *FluidMap* and Voronoi-MW$^{\alpha}$ for single points, slightly favouring the latter. E2 noticed that such small islands can be interpreted as outliers and thus support outlier detection, a task not directly evaluated in the study.

T3: Next, we asked the experts to estimate the distribution of attribute categories. Estimation errors arise from two sources: (1) The visualisation method itself may introduce inaccuracies by failing to preserve the true attribute distribution (AD) in its visual distribution (VD). (2) Experts may misinterpret or misjudge the relative area associated with an attribute category. In Table 1, we numerically evaluated the methods with respect to quantity preservation. A comparison between the actual and visual attribute distributions along the experts' estimations is provided in the supplementary material. While Voronoi generates a clear and spatially consistent separation of the visual plane supporting **T1**, it severely over- or under-represents attribute categories, inducing incorrect conclusions. For instance, experts inferred that over half of the Titanic's passengers survived with evenly distributed genders, whereas the actual distribution reveals different proportions. While Voronoi-MW$^{\alpha}$ creates a more faithful visual distribution, it still fails to represent weights accurately, such that experts misjudge attribute distributions. Only *FluidMap* and Nmap preserve attribute distributions reliably. E5 noted that islands in between larger areas made the tasks difficult because he was forced to mentally subtract the smaller islands from the larger surrounding area. He argued that these islands convey a misleading perception of the surrounding area's size due to the Gestalt Principle of Closure [Tod08], which states that humans mentally close gaps. This can be an issue for all methods, especially for multi-valued attributes (e.g., `Class` and `Property Type Purchased`). Most experts (except E4) found the rectangular shape of Nmap helpful for estimating proportions, provided the areas were not too slim or fragmented. E4 instead preferred rounded, blob-like shapes, as in *FluidMap* and Voronoi-MW$^{\alpha}$, over the others, describing them as

more intuitive and also suitable to estimate attribute distributions. Regardless of shape preferences, all experts performed best with *FluidMap* and Nmap, consistent with their faithful quantity preservation by design.

T4: In the final task, the experts were asked to judge which attributes most strongly influence the arrangement of data points in the projection. This more complex topology-related task revealed disagreement among experts. E5 argued that only methods creating clear category boundaries that divide the visual plane, as Voronoi (e.g., `Sex`), allow reliable conclusions about the attributes' influence on the projection topology. E3, in contrast, argued that the task is also solvable using methods like *FluidMap*, where different coloured areas are placed at opposite sides of the plot, corresponding to areas where no data point with the respective attribute is located. He found that the colouring of these areas was inconsistent, making the task a bit more challenging. He still reached similar conclusions as with Voronoi, for example, that `Survived` and `Sex` strongly influence the topology, while `Class`, being most fragmented, seems to have the least effect. E4 perceived no major differences between the methods at all, claiming that attribute categories well separated by Voronoi were still well separated in *FluidMap*, since no data points are included in the area fragments spreading into conflicting attribute areas.

General comments: The flexibility of the shapes in *FluidMap* had mixed connotations: Initially, irregular shapes, especially thin extensions or offshoots (e.g., `Second Class`), were perceived as unaesthetic and mentally demanding for distribution estimation. Some questioned whether such irregularities were artifacts or signs of greater accuracy. In direct comparison, the rounded or edged shapes of the Voronoi-based approaches conveyed a more pleasant appearance. Over the course of the study, participants developed a growing appreciation for the irregular shapes, along with increased confidence in *FluidMap*, recognising both its reliable task performance and the strong memorability of the generated shapes. E1 positively highlighted the colour blob shapes and suggested post-processing to reduce offsprings, aligning with E2 and E3's wish for clearer edges and E4's preference of shapes generated by Voronoi-MW$^{\alpha}$ over those of *FluidMap*. E2–E5 also expressed a preference for connected areas of the same colours, especially for attribute distribution and topology-related tasks. Connectedness itself, however, is no direct optimisation criterion for any of the methods under observation and depends largely on data placement and attribute distribution. E5 concluded that *FluidMap* is the most convincing method overall, as it serves as a compromise between the others. Even though he initially preferred Voronoi-MW$^{\alpha}$ for the classification tasks, rectangular shapes as in Nmap for the attribute distribution estimation, and Voronoi for the topology-related task, he was positively surprised by how well he was able to successfully solve *all* given tasks with *FluidMap* despite his initial skepticism. This final thought aligns with the statements of the other experts.

5. Discussion and Future Work

While *FluidMap* and Voronoi-MW$^{\alpha}$ share a similar distance–weight formulation, *FluidMap* differs fundamentally in its iterative, region-growing process, where only already-reached pixels are considered. This enables a controlled expansion of areas around data points, ensuring quantity preservation and spatial consistency, as opposed to the global assignment in Voronoi-based methods. Regarding algorithmic requirements, *FluidMap* produces deterministic results for given input parameters similar to its competitors. Given its iterative nature, it is slower in runtime. However, the main difference between *FluidMap* and its competitors is that it directly incorporates data point attributes into the assignment process. When areas of the same attribute connect, their points are grouped under a common label. Pixels are still assigned to individual data points, but within a cooperating group, no reassignments are considered. Since pixels assigned to data points of the same attribute are rendered with the same colour, this simplifies and speeds up the assignment process. A side effect of this is that, unlike attribute-independent assignments, as in Voronoi-based and Nmap methods, pixel-to-point assignments may vary across attributes of the same dataset, possibly generating slightly different visualizations (e.g., Figures 4 and 5). Although the pressure dynamics remain unchanged, the group formation depending on the attribute used for enrichment differs, which explains the variations. Yet, domain experts did not view this as a drawback. For cross-attribute consistency, that is, assignment independent of the chosen attribute, the algorithm can be adjusted by omitting the concept of cooperating attribute groups, which coincidentally eliminates the need for recomputation. A limitation of *FluidMap* is the possibility of thin pixel bridges. An example can be seen in the layout of the `Class` attribute in Figure 4, where the area of the green data point at the top winds its way to the lower left corner via a thin bridge. To suppress these offshoots and address the experts' wishes for smoother shapes, one could introduce a surface tension of the fluids, restricting the flexibility of area shapes. In future work, our technique could be further optimised in terms of runtime and extended, for instance, by incorporating attraction forces between fluids of the same attribute category, thereby improving the connectedness of attributes. Another interesting extension might be to integrate temporal dynamics for evolving datasets.

For the layout enrichment to faithfully represent the scatterplot, we introduced condition **C3**, that is, the position of the projected data points and the visualisation plane Ω are assumed to be fixed. While fixing the points' position in 2D is crucial to faithfully represent the projected data, the borders of Ω offer some range of flexibility as they are not determined by the projection nor explicitly defined by the underlying data. While the borders shouldn't lie too far from the projected points to efficiently use the full size of the visualisation screen, adjusting them to a certain degree could help our method to limit areas spreading to spaces far from the projected points (e.g., Blobs), thereby improving distance and neighbourhood preservation. One possible solution might be, for example, to adjust the borders such that the centre of Ω corresponds to the weighted centroid of the projected data points. Furthermore, *FluidMap* is designed to be space-filling, that is, it leaves no unassigned white spaces in the visualisation's background. Depending on the size of the dataset and the density distribution over the spatial layout, it may be beneficial to allow white spaces in relatively unpopulated areas of the plot to avoid creating incorrect perceptions for these regions. This is closely related to limitations stemming from overplotting of data points. This is not an issue unique to *FluidMap*; it occurs in all exclusive layout enrichment methods under consideration. Introducing white spaces can have a positive effect on scalability in terms of different dataset sizes and densities, and support a more

universal applicability of the layout enrichment. Closely related are problems arising from very dense clusters. Here, our initial assignment of source pixels to data points is comparable to the point-to-pixel mapping proposed in [RGE19]. To ensure spatial consistency, that is, avoid remapping, while maintaining efficiency, *FluidMap* could be adapted to use a high pixel resolution in areas with dense clusters and switch to a coarser resolution in sparsely populated regions.

General limitations of layout enrichments, such as the visual restriction to the number of different attribute values that can be displayed by colour, persist due to the limited number of colours that can be distinguished from one another. To the best of our knowledge, there does not exist an in-depth survey that compares existing layout enrichment techniques apart from the subsection in [NA19]. Further, we miss an exhaustive list of quality measures to evaluate the methods. Our proposed metrics offer a starting point, but could be further refined and extended. We also miss a user study for background enrichments that investigates, for instance, aesthetic criteria. This may involve the connectivity of areas with the same attribute or the shape of class boundaries. Further research is needed to study the perception of relative quantities when dealing with areas of various shapes and colour and their impact on cognitive load. A restriction on the shape, such as the aspect ratio of rectangles in Nmap, could facilitate a more accurate size comparison between assigned areas. **T3** of our expert study could serve as a starting point for a larger-scale user study. For creating layout enrichments in general, the user is usually not interested in optimising a single quality criterion, such as distance or quantity preservation. Thus, future work should continue to focus on the combination or trade-off between quality measures.

6. Conclusion

In this paper, we introduced *FluidMap*, a novel approach to space-filling layout enrichment for scatterplot projections inspired by principles of fluid dynamics. By addressing key limitations of existing methods such as over- or underrepresentation of categorical attributes in Voronoi partitioning, spatial inconsistency in Nmap, and a mixture of both in the proposed Voronoi-MW$^{\alpha}$, *FluidMap* provides an innovative solution that ensures proportional representation of data weights while maintaining spatial consistency and flexibility in shape generation. Through qualitative and quantitative evaluations, we demonstrated that *FluidMap* enhances the interpretability of scatterplot backgrounds, allowing users to better discern data relationships and distributions at a glance. Its ability to generate contextually meaningful layouts positions it as a valuable tool for visualisation practitioners seeking to enrich scatterplots while preserving the fidelity of underlying data representations. We believe *FluidMap* sets the groundwork for more flexible, data-driven layout enrichment techniques, contributing to the broader effort of creating more intuitive and effective visualisation tools.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161 (Projects A01 and A03).

Open access funding enabled and organized by Projekt DEAL.

Ethics Statement

Ethical approval is not applicable for this study as it did not involve human or animal subjects.

References

[AE84] Aurenhammer F., Edelsbrunner H.: An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recognition 17*, 2 (1984), 251–257. https://doi.org/10.1016/0031-3203(84)90064-5.

[Aup07] Aupetit M.: Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing 70*, 7-9 (2007), 1304–1330. https://doi.org/10.1016/J.NEUCOM.2006.11.018.

[Aur87] Aurenhammer F.: Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing 16*, 1 (1987), 78–96. https://doi.org/10.1137/0216006.

[Aur91] Aurenhammer F.: Voronoi diagrams - A survey of a fundamental geometric data structure. *ACM Computing Surveys 23*, 3 (1991), 345–405. https://doi.org/10.1145/116873.116880.

[BDL05] Balzer M., Deussen O., Lewerentz C.: Voronoi treemaps for the visualization of software metrics. In *Proceedings of the 2005 ACM Symposium on Software Visualization* (New York, NY, USA, 2005), SoftVis '05, Association for Computing Machinery, p. 165–172. https://doi.org/10.1145/1056018.1056041.

[BTB13] Broeksema B., Telea A. C., Baudel T.: Visual analysis of multi-dimensional categorical data sets. *Computer Graphics Forum 32*, 8 (2013), 158–169. https://doi.org/10.1111/CGF.12194.

[BWT*24] Blumberg D., Wang Y., Telea A., Keim D. A., Dennig F. L.: Inverting multidimensional scaling projections using data point multilateration. In *16th Int. EuroVis Workshop Visual Analytics* (2024), The Eurographics Association. https://doi.org/10.2312/eurova.20241112.

[CGSQ11] Cao N., Gotz D., Sun J., Qu H.: DICON: Interactive visual analysis of multidimensional clusters. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2581–2590. https://doi.org/10.1109/TVCG.2011.188.

[CLNL87] Carr D. B., Littlefield R. J., Nicholson W. L., Littlefield J. S.: Scatterplot matrix techniques for large n. *Journal of the American Statistical Association 82*, 398 (1987), 424–436. URL: http://www.jstor.org/stable/2289444.

[CLRP13] Choo J., Lee C., Reddy C. K., Park H.: UTOPIAN: user-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 1992–2001. https://doi.org/10.1109/TVCG.2013.212.

[Daw95] Dawson R. J. M.: The "unusual episode" data revisited, 1995. http://jse.amstat.org/v3n3/datasets.dawson.html, last

accessed 2025-11-01. https://doi.org/10.1080/10691898.1995.11910499

[DJP*24] Dennig F. L., Joos L., Paetzold P., Blumberg D., Deussen O., Keim D. A., Fischer M. T.: The categorical data map: A multidimensional scaling-based approach. In *Proceedings of the 2024 IEEE Visualization in Data Science (VDS)* (2024). https://doi.org/10.1109/VDS63897.2024.00008.

[DSF*14] Duarte F. S. L. G., Sikansi F., Fatore F. M., Fadel S. G., Paulovich F. V.: Nmap: A novel neighborhood preservation space-filling algorithm. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2063–2071. https://doi.org/10.1109/TVCG.2014.2346276.

[dSRM*15] da Silva R. R. O., Rauber P. E., Martins R. M., Minghim R., Telea A. C.: Attribute-based visual explanation of multidimensional projections. In *6th International EuroVis Workshop on Visual Analytics (EuroVA)* (2015), Eurographics Association, pp. 31–35. https://doi.org/10.2312/EUROVA.20151100.

[For86] Fortune S.: A sweepline algorithm for voronoi diagrams. In *Proceedings of the Second Annual Symposium on Computational Geometry* (1986), pp. 313–322. https://doi.org/10.1007/BF01840357.

[HFA17] Heulot N., Aupetit M., Fekete J.-D.: ProxiViz: an interactive visualization technique to overcome multidimensional scaling artifacts. *Proceedings of IEEE InfoVis, poster 2* (2012).

[HHS20] Hogräfer M., Heitzler M., Schulz H.-J.: The state of the art in map-like visualization. *Computer Graphics Forum 39*, 3 (2020), 647–674. https://doi.org/10.1111/cgf.14031.

[JHM*13] Janetzko H., Hao M. C., Mittelstädt S., Dayal U., Keim D. A.: Enhancing scatter plots using ellipsoid pixel placement and shading. In *46th Hawaii International Conference on System Sciences, HICSS 2013, Wailea, HI, USA, January 7-10, 2013* (2013), IEEE Computer Society, pp. 1522–1531. https://doi.org/10.1109/HICSS.2013.197.

[Jol86] Jolliffe I. T.: *Principal Component Analysis*. Springer, 1986. https://doi.org/10.1007/978-1-4757-1904-8.

[JS91] Johnson B., Shneiderman B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceeding Visualization '91* (1991), pp. 284–291. https://doi.org/10.1109/VISUAL.1991.175815.

[Kru64] Kruskal J. B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (1964), 1–27. https://doi.org/10.1007/BF02289565.

[KSDK11] Koh L. C., Slingsby A., Dykes J., Kam T. S.: Developing and applying a user-centered model for the design and implementation of information visualization tools. In *15th International Conference on Information Visualisation* (2011), pp. 90–95. https://doi.org/10.1109/IV.2011.32.

[LA11] Lespinats S., Aupetit M.: CheckViz: Sanity check and topological clues for linear and non-linear mappings. *Computer Graphics Forum 30*, 1 (2011), 113–125. https://doi.org/10.1111/J.1467-8659.2010.01835.X.

[MBC*23] Morariu C., Bibal A., Cutura R., Frénay B., Sedlmair M.: Predicting user preferences of dimensionality reduction embedding quality. *IEEE Transactions on Visualization and Computer Graphics 29*, 1 (2023), 745–755. https://doi.org/10.1109/TVCG.2022.3209449.

[MCMT14] Martins R. M., Coimbra D. B., Minghim R., Telea A. C.: Visual analysis of dimensionality reduction quality for parameterized projections. *Computer Graphics 41* (2014), 26–42. https://doi.org/10.1016/J.CAG.2014.01.006.

[MG13] Mayorga A., Gleicher M.: Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics 19*, 9 (2013), 1526–1538. https://doi.org/10.1109/TVCG.2013.65.

[MGU*21] Müller J., Garrison L. A., Ulbrich P., Schreiber S., Bruckner S., Hauser H., Oeltze-Jafra S.: Integrated dual analysis of quantitative and qualitative high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics 27*, 6 (2021), 2953–2966. https://doi.org/10.1109/TVCG.2021.3056424.

[MMT15] Martins R. M., Minghim R., Telea A. C.: Explaining neighborhood preservation for multidimensional projections. In *Computer Graphics and Visual Computing, CGVC 2015, London, United Kingdom, September 16-17, 2015* (2015), Borgo R., Turkay C., (Eds.), Eurographics Association, pp. 7–14. https://doi.org/10.2312/CGVC.20151234.

[MT20] Mu W., Tong D.: Distance in Spatial Analysis: Measurement, Bias, and Alternatives. *Geographical Analysis 52*, 4 (2020), 511–536. https://doi.org/10.1111/gean.12254.

[NA19] Nonato L. G., Aupetit M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics 25*, 8 (2019), 2650–2673. https://doi.org/10.1109/TVCG.2018.2846735.

[RGE19] Raidou R. G., Groller M. E., Eisemann M.: Relaxing Dense Scatter Plots with Pixel-Based Mappings. *IEEE Transactions on Visualization and Computer Graphics 25*, 6 (2019), 2205–2216. https://doi.org/10.1109/TVCG.2019.2903956.

[RML24] Rave H., Molchanov V., Linsen L.: Uniform sample distribution in scatterplots via sector-based transformation. In *2024 IEEE Visualization and Visual Analytics (VIS)* (2024), pp. 156–160. https://doi.org/10.1109/VIS55277.2024.00039.

[Shn92] Shneiderman B.: Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics 11*, 1 (1992), 92–99. https://doi.org/10.1145/102377.115768.

[SLD20] Scheibel W., Limberger D., Döllner J.: Survey of treemap layout algorithms. In *Proceedings of the 13th International Symposium on Visual Information Communication and Interaction* (New York, NY, USA, 2020), VINCI '20, Association for Computing Machinery. https://doi.org/10.1145/3430036.3430041.

[Ste03] Stepin M.: hqx - high-quality magnification for low-resolution images. Online, 2003, last accessed 2025-11-01. Available at https://web.archive.org/web/20070717064839/www.hiend3d.com/hq4x.html.

[SvLB10] Schreck T., von Landesberger T., Bremm S.: Techniques for precision-based visual analysis of projected data. *Information Visualization 9*, 3 (2010), 181–193. https://doi.org/10.1057/IVS.2010.2.

[Tod08] Todorovic D.: Gestalt principles. *Scholarpedia 3*, 12 (2008), 5345. https://doi.org/10.4249/SCHOLARPEDIA.5345.

[Tuf92] Tufte E. R.: *The visual display of quantitative information*. Graphics Press, 1992. https://doi.org/10.2307/530384.

[Ves99] Vesanto J.: Som-based data visualization methods. *Intell. Data Anal. 3*, 2 (1999), 111–126. https://doi.org/10.1016/S1088-467X(99)00013-X.

[Wat05] Wattenberg M.: A note on space-filling visualizations and space-filling curves. In *Information Visualization, IEEE Symposium on* (Los Alamitos, CA, USA, oct 2005), IEEE Computer Society, pp. 181–186. https://doi.org/10.1109/INFVIS.2005.1532145.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information