Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment

Luis Gustavo Nonato, *Member, IEEE*, and Michaël Aupetit

Abstract— Visual analysis of multidimensional data requires expressive and effective ways to reduce data dimensionality to encode them visually. Multidimensional projections (MDP) figure among the most important visualization techniques in this context, transforming multidimensional data into scatter plots whose visual patterns reflect some notion of similarity in the original data. However, MDP come with distortions that make these visual patterns not trustworthy, hindering users to infer actual data characteristics. Moreover, the patterns present in the scatter plots might not be enough to allow a clear understanding of multidimensional data, motivating the development of layout enrichment methodologies to operate together with MDP. This survey attempts to cover the main aspects of MDP as a visualization and visual analytic tool. It provides detailed analysis and taxonomies as to the organization of MDP techniques according to their main properties and traits, discussing the impact of such properties for visual perception and other human factors. The survey also approaches the different types of distortions that can result from MDP mappings and it overviews existing mechanisms to quantitatively evaluate such distortions. A qualitative analysis of the impact of distortions on the different analytic tasks performed by users when exploring multidimensional data through MDP is also presented. Guidelines for choosing the best MDP for an intended task are also provided as a result of this analysis. Finally, layout enrichment schemes to debunk MDP distortions and/or reveal relevant information not directly inferable from the scatter plot are reviewed and discussed in the light of new taxonomies. We conclude the survey providing future research axes to fill discovered gaps in this domain.

Index Terms— Multidimensional Projection, Dimensionality Reduction, Multidimensional Scaling, Error Analysis, Layout Enrichment.

1 Introduction

Multidimensional scaling (MDS) has long been employed as a mechanism to embed data instances into a Cartesian space. In the context of visualization, where the embedding space is typically two or three-dimensional, MDS has been called *Multidimensional Projection* (MDP) [73, 88, 152], being characterized as the family of mappings capable of producing similarity-preserving point-based (scatter plot) layouts. The precise meaning of "similarity" depends on the data type and associated relations, ranging from an ordering of instances to distances in high-dimensional spaces.

The capability of preserving similarity has rendered MDP methods fundamental components of many visualization applications, fostering a multitude of developments. Such developments go beyond the proposal of new MDP methods, extending to novel methodologies for the visual analysis of distortions introduced during the mapping process, and the design of new visual metaphors to further assist the exploration and understanding of MDP layouts.

Existing surveys [29, 106, 159, 173] and books [23, 83] devoted to overview the multitude of MDS/MDP methods do not provide a comprehensive review of all the recent achievements leveraged by MDP methods, mainly in the context of visualization. In fact, existing reports focus mainly on mathematical and computational aspects of the mapping process [29, 106], on quantitative comparison of particular MDP methods [118, 159], or on interaction tasks involving MDP layouts [131], disregarding important efforts towards rendering MDP a trustworthy and highly informative visualization mechanism.

Focus of this survey The present survey reviews MDP methods from such a broader perspective, covering not only recent advances in terms of mapping mechanisms, but also related methodologies for the visual analysis and interpretation of MDP layouts. This survey encompasses four main facets of MDP in the context of visualization:

- MDP methods and their particular properties;
- MDP types of errors and ways to measure them;
- Visual Analytic tasks that rely on MDP scatter plots and the impact of errors onto such tasks;
- Methodologies for enriching MDP layouts so as to visually identify distortions and errors as well as further facilitate data exploration and understanding.

An outcome of such comprehensive discussion is the identification of open issues in different fronts, opening new research possibilities.

Techniques such as SPLOM [98] and star plots [44, 87] are out of the scope of this survey. Although those families of methods also produce point-base layouts, they are not designed to directly preserve similarity between instances, even though some variants bear such property [88]. Another closely related topic that will not be approached in this review is graph drawing [151]. Algorithms such as force-directed placement can be employed for both MDP and graph drawing when adjacency matrices are interpreted as similarity matrices, however, graph drawing has particularities such as preventing edge crossings, which are not an issue in the context of MDS/MDP, justifying our decision for not including them in this survey.

Outline In order to facilitate the reading, we organize the survey in five self contained sections. Section 2 reviews existing MDP methods according to distinct taxonomies, discussing specific properties and their impact in the context of visualization. In particular, Section 2.4 discusses which MDP to use given visualization factors. Distortions and errors intrinsically present in MDP layouts and quantitative metrics to gauge them are described in Section 3, also discussing in Section 3.4, how MDP distortions may impact visual analysis. Section 4 provides an overview of typical user tasks where MDP methods play a fundamental role, including discussions about which tasks require careful map interpretation (Section 4.2) and which MDP best support the intended task (Section 4.3). Methodologies for "enriching" MDP layouts so as to ease distortion analysis and data interpretation are described in Section 5, with a discussion on how to choose a proper layout enrichment scheme (Section 5.3). We conclude the survey in Section 6, presenting a set of open questions and aspects that deserve further investigation.

- *Nonato is with University of São Paulo and New York Universiy. E-mail: gnonato@icmc.usp.br*
- *Aupetit is with Qatar Computing Research Institute at Hamad Bin Khalifa University. E-mail: maupetit@hbku.edu.qa*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx/

2 MULTIDIMENSIONAL PROJECTION FOR VISUALIZATION

In this section we provide an overview of MDP techniques that have been employed in visualization applications. Our intent is not only to provide a comprehensive survey of MDP methods, but also discuss characteristics and particularities of MDP techniques that are important in the context of visualization. Before discussing the interplay between MDP technical characteristics and their impact on visualization (2.4), we first present the relation between MDP and visual perception factors (2.1), the mathematical notation (2.2), and the proposed taxonomies with a discussion about their properties and how existing MDP methods match those taxonomies (2.3).

2.1 MDP and Visual Perception Factors

Bertin [19] showed experimentally that brightness, size, orientation and position are graphical variables best suited to encode ordered elements, position being the most effective, while hue and shape are the most suited to encode categorical (unordered) content. MDP scatter plots rely on spatialization to encode similarities, mapping each data item to a point on the visual space such that the relative pairwise proximities reflect at best the corresponding pairwise similarities. As recommended by Bertin, similarity, the most important information in MDP, is encoded by spatial proximity, the primary graphical variable.

The Gestalt laws of proximity and similarity [167] are two pre-attentive detection processes of the human visual perception system. The *law of proximity* states that groups of points *spatially close to each other* are pre-attentively (i.e. almost instantly and without cognitive effort) perceived to share a common set of abstract features not visualized *per se*. The *law of similarity* states that items whose *markers have the same appearance* (color or shape) are pre-attentively perceived to share also some abstract similarity. Based on the law of proximity, visual grouping of points in MDP scatter plots would be instantaneously perceived as groups of similar items in the data space. Thus, it seems important to ensure that most of the spatial proximities match the items' similarities to benefit from this pre-attentive law of perception and increase effectiveness of a MDP result.

2.2 Notation and Nomenclature

In a MDP setting, data usually come as either:

- a data table $\mathbf{X}$ with identified features (attributes) associated to each data instances,
- a matrix Δ where each entry accounts for the (dis)similarity between items.

Several authors have tried to formalize the concepts of similarity and dissimilarity [92, 133, 139], but a standard definition is yet to be found. To avoid any controversy, we opt to provide an intuitive definition of similarity and dissimilarity. Intuitively, a *dissimilarity* is a numerical measure of how different two data items are, the larger the measure the more dissimilar the items are. A particularly important example of dissimilarity measure is the Euclidean distance. Conversely, a *similarity* measure gauges how alike two data items are, where high similarity values encode nearly identical items while low similarity means the items are different. A less common setting is to provide an ordering for the items, meaning that nearly consecutive items are more similar.

We denote $\mathbf{x}_i = (x_i^{[1]} \dots x_i^{[p]})$ the position of data instance i in the real (feature) space $\mathscr{D}_x = \mathbb{R}^p$, where $\mathbf{X} = (\mathbf{x}_i)_{1\dots n}$ is the data matrix. We denote $\boldsymbol{\delta}_i = (\delta_{i1} \dots \delta_{in})$ the position of instance i in the distance (or dissimilarity) space $\mathscr{D}_\delta = (\mathbb{R}^+)^n$, with $\delta_{ij} \in \mathbb{R}^+$ the dissimilarity between data i and j and $\Delta = (\delta_{ij})_{(1\dots n,1\dots n)}$. $\mathscr{D}$ denotes the data space and it can be $\mathscr{D}_x$ or $\mathscr{D}_\delta$ depending on the original data format. Instances are mapped to a visual space by the transformation $\Phi : \mathscr{D} \to \mathscr{M}$, giving rise to points $\mathbf{y}_i = (y_i^{[1]} \dots y_i^{[q]})$ that correspond to the position of item i in the visual space $\mathscr{M}$, where $q \in \{2,3\}$. $d_{ij} \in \mathbb{R}^+$ denotes the distance between points i and j in the visual space and $D = (d_{ij})_{(1\dots n,1\dots n)}$ is the distance matrix therein. d_{ij} aims to provide a representation of δ_{ij} in the visual space, thus visually conveying the similarity relation of the original data items. Typically, d_{ij} is assumed to be the Euclidean distance: $d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|$. We can consider a set of r data instances as *landmarks* used as reference points used to map the other data, mainly to lessen computational burden. Landmarks can also be used as control points to steer the projection. To make clear the context, we will use the expression *control points* to refer to steerable MDP and *landmarks* (or pivots) as reference points for interpolation purposes, that is, landmarks are not supposed to be (interactively) displaced.

In our context, a *cluster* or a *class* is formally a group of data items. Both differ conceptually though: a *cluster* is always defined with respect to a similarity measure or a metric space, while a *class* can be stand-alone, an input taken for granted. A cluster can be formed by items perceived as nearby each other in the visual space $\mathscr{M}$, or by running an automatic clustering algorithm in the data space $\mathscr{D}$. Distinctively, a class is formed by items sharing some given or discovered semantic (*e.g.* "spam" or "malignant cell"). Typically in MDP context, clusters are identified based on point location in $\mathscr{M}$ and visually compared to pre-existing color-coded classes, a task defined as *Match Clusters and Classes* by Brehmer et al. [26], or their content can be analyzed and annotated to generate a new class, the *Name cluster* task from Brehmer et al. [26]. The interplay between clusters and classes has been analyzed by Aupetit [5] to study how MDP are evaluated visually, and to serve as design rationale for ClassiMap [90]. In the sequel, we denote clusters and classes as $\mathscr{C} = \{c_1 \dots c_K\}$ and $\mathscr{L} = \{l_1 \dots l_L\}$, respectively, that is, c_i (l_i) represents the instances belonging to the ith cluster (class).

2.3 MDP Taxonomies

The provided taxonomies aim to characterize the multiple facets of MDP techniques. Some of those taxonomies have already been proposed in the literature, organizing MDP methods according to the their mathematical formulation [118], capability of interaction [33], sparsity of the problem [83], and ability to locally preserve the geometry of the data [59]. The proposed taxonomies are more comprehensive, encompassing several other traits, as summarized in the columns of Table 1.

Specifically, we categorize MDP methods as to *Data Types*, *Linearity*, flexibility for *Supervision*, capability for dealing with *Multi-level* structures, *Locality*, *Steerability*, *Stability*, and capability of handling *Out-of-Core* (OOC) data. The meaning of each taxonomy and corresponding properties will be clear throughout this section.

Techniques in the first column of Table 1 are "core" techniques in the sense their mathematical foundation and/or the manner they deal with neighborhood structures differ considerably from one to another. A multitude of variants of those techniques have been proposed in the literature, some of which are discussed in the following subsections and summarized in Table 2. In Table 1, MDP techniques are listed in chronological order, making clear that most of the recent developments (rows colored in blue) have been fostered by visualization applications, which have leveraged the creation of methods with particular properties and functionalities.

The check mark symbol ✔ in Table 1 indicates the method intrinsically handles the pointed data type or holds the marked property, while the dot symbol • implies the method can straightforwardly be adapted to hold the property or that there are variants in the literature bearing the corresponding property (or data type). Empty dots ◦ indicate the method can hold the property only under special conditions. Besides characterizing MDP techniques and their properties, we also discuss computational aspects.

2.3.1 MDP and Data Types

Data types commonly handled by visualization applications can be grouped into five major categories: Dissimilarity (Di), Ordinal (Or), Cartesian (Ca), Neighborhood (Ne), and Categorical (Ct). As discussed in Section 2.2, those data types are given as a data matrix $\mathbf{X}$ or a dissimilarity matrix Δ. Some techniques are more flexible with respect to the types of data they are able to deal with, but an important fact that shows up from Table 1 is that there is no MDP method capable of handling data from all the five major categories.

Table 1: MDP methods vs. Taxonomies described in Section 2.3. The check mark ✔ indicates the method naturally bears the property or can handle the type of data in the corresponding column. The symbol • implies the method can easily be adapted to hold the property or there are variants in the literature bearing the corresponding property (or handling the data type). The empty circle symbol ○ indicates the method can satisfy the property under special conditions. **Data Types**: type of data that MDS methods can handle - dissimilarity (Di), ordinal (Or), Cartesian (Ca), Neighborhood Structures (Ne), Categorical (Ct). The complexity column (Comp.) shows the computation burden with respect to the number of iterations (i); prototypes (l), basis functions (m), instances (n), dimensions (p), and landmarks (r).

Technique	Taxonomy											
	Data Types					Linearity	Supervision	Multi-level	Locality	Steerability	Stability	OOC	Comp.
	Di	Or	Ca	Ne	Ct								
PCA [65]			✔			✔					•	✔	$O(p^3)$
LDA [50]			✔			✔	✔					✔	$O(np^2+p^3)$
Classical MDS [155]	✔		✔	•		✔	•			•	•		$O(n^3)$
Kruskal [79]	✔	✔	✔	•			•			•		•	$O(in^2)$
NLM [132]	✔		✔	•			•			•		•	$O(in^2)$
MCA [17]			•		✔								$O(n^3)$
Smacof [42]	✔	✔	✔	•			•					•	$O(in^2)$
SOM [126]			✔									✔	$O(l^2np+l^2)$
FastMap [48]	✔		✔	•			•					•	$O(n)$
Chalmers [32]	✔		✔	•			•	•		•	○	✔	$O(in)$
GTM [22]			✔									•	$O((lnp)^3+nr^3)$
Pekalska [120]	✔		✔	•		✔	•						$O(r^3+rn)$
CCA [43]	✔		✔	•			•					✔	$O(l^2)$
LLE [129]	✔		✔	•			•		✔			•	$O(n^3)$
Isomap [153]	✔		✔	✔			•					•	$O(n^3)$
Lapl. Eigenmaps [15]	✔		✔	✔			•		✔			•	$O(n^3)$
Force-Directed [152]	✔		✔				•	•				✔	$O(in^2)$
LTSA [180]			✔			•			✔				$O(n^3)$
MVU [169]	✔		✔	•			•		✔			•	$O(n^3)$
LSP [117]	✔		✔	✔			•	•	✔	✔	○		$O(n^3)$
SNE [64]	✔		✔	•			•	•		○		•	$O(in^2)$
PLMP [118]			✔			•	•	•	✔	✔	○		$O(n^3)$
LAMP [73]			✔				•	•	✔	✔	✔	✔	$O(pn)$
RBF-MP [2]	✔		✔	•			•			✔	○	•	$O(r^3+n)$
LoCH [47]	✔		✔	•			•		✔	✔		✔	$O(n\sqrt{(n)})$
ClassiMap [90]	✔		✔	•			✔					•	$O(in^2)$
Kelp [13]	✔		✔	•		•	•		✔	✔	○	•	$O(r^3)$

Table 2: Variants of methods listed in Table 1.

Core Method	Variants
PCA [65]	see [181] for variants; iPCA [71], PCP [182]
LDA [50]	see [61] for variants
Classical MDS [155]	L-MDS [144], Pivot-MDS [24], CFMDS [113]
MCA [17]	see [154] for variants
SOM [126]	see [150] for variants
Chalmers [32]	Glimmer [70]
LLE [129]	SLLE [178]
Isomap [153]	S-Isomap [53]
LTSA [180]	LLTSA [179]
MVU [169]	MUHSIC [147]
LSP [117]	PLP [115], E-LSP [33], Hipp [116]
SNE [64]	NeRV [162], t-SNE [158], DS t-SNE [77], Q-SNE [69] A-tSNE [122], H-SNE [121], BH-tSNE [157]

"Data Types" columns in Table 1 show that most MDP techniques can handle data given as Δ and that all of them can obviously deal with Cartesian data, as Cartesian data can be converted to dissimilarities by simply applying a metric such as Euclidean distance. In contrast, not all methods able to process Cartesian data are capable of handling dissimilarities. MDP methods able to handle dissimilarities can also be adapted to operate with neighborhood information, that is, when the only known information is the list of neighbors of each instance. Neighbor information allows for building a neighborhood graph where each instance is connected to its neighbors. Dissimilarities can then be derived by considering the length of the shortest path between each pair of instances, the shorter the path the more similar the instances are. Therefore, MDP techniques able to handle dissimilarities are, in general, the most versatile ones, an important trait for visualization purposes.

Ordinal data gives an order relation between pairs of instances, that is, if δ_{ij} and δ_{kl} correspond to the dissimilarities between $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$, $\mathbf{x}_l$ respectively, then the ordinal relation ensures that either $\delta_{ij} \leq \delta_{kl}$ or $\delta_{ij} \geq \delta_{kl}$. In less mathematical terms, distances are not so important, their ordering is what really matters. The mapping should be such that $\delta_{ij} \leq \delta_{kl}$ implies $d_{ij} \leq d_{kl}$. Notice that dissimilarity values might be unknown, only the order relation needs to be provided. Ordinal data typically appears in applications where pairs of instances are ranked according to some criteria (no dissimilarity values are known), as for example the position of countries according to their HDI. As Table 1 shows, only Kruskal [79] and Smacof [42] methods are able to handle oridinal data type.

Measuring the dissimilarity between instances whose attributes are given as labels of certain categories is not an easy (or even doable) task, justifying the small number of MDP techniques able to handle categorical data. MCA [17] and its variants [154] are among the few methods capable of mapping categorical data to a visual space such that closeness reflects similarities, e.g., statistical properties of the attributes. One could argue that categorical information can be numerically encoded by assigning a value to each element of a category (dog = 0, cat = 1, etc.), thereby allowing the computation of distances. The problem with such approach is that the resulting (dis)similarity depends on how the values are assigned to the elements of each category. For example, two instances sharing four out of five attributes can be more distant than instances sharing only one attribute depending on how the values are assigned to the elements of each category.

2.3.2 Linear *vs* Nonlinear Methods

Mathematically speaking, linear MDP methods map data to a Cartesian (visual) space using a single transformation $\Phi : \mathscr{D} \rightarrow \mathscr{M}$ that satisfies the condition $\Phi(au+bv) = a\Phi(u)+b\Phi(v)$. As shown in Table 1 ("Linearity" column), there are a few MDP techniques that are truly linear in strict mathematical terms, although the word "linear" shows up often as part of the name of several MDP methods, causing some confusion, mainly for non-experienced users. Moreover, several non-linear MDP techniques rely on linear mechanisms to map instances to a visual space. For instance, LLE [129], LTSA [180], and Laplacian Eigenmaps [15] accomplish the mapping based on eigen-decomposition of particular matrices. LSP [117] and PLP [115] rely on Laplacian linear system solvers to perform the transformation to the visual space. Both eigen-decomposition and Laplacian systems are well known linear mechanisms, however, the mapping Φ resulting from those methods are not linear at all, that is, they do not satisfy the condition $\Phi(au+bv) = a\Phi(u)+b\Phi(v)$. There are methods such as the one proposed by Pekalska [120] that performs a linear mapping from the space of dissimilarities ($\mathscr{D}_\delta$) to a Cartesian space, however, data instances themselves are not mapped linearly.

Linear methods have the advantage of being computationally efficient, because, once the linear transformation is computed, the mapping of each instance to the Cartesian visual space is accomplished through a simple matrix-vector multiplication. However, linear methods can not unfold complex structures onto a visual space, making them prone to introduce considerable distortions when dealing with "tangled" manifolds, what is a serious issue for visualization purposes. Although alternatives such as local alignment [179] and steerability [118] have been proposed to mitigate the issue of unfolding, linear methods can rarely outperform non-linear techniques in that aspect.

The kernel trick has long been used as a mechanism to render linear methods able to handle non-linear structures [21]. In the context of MDP, the kernel trick has been employed to empower linear techniques such as PCA [134] and classical MDS [170], being also the foundation of MDP techniques such as Kelp [13].

2.3.3 Capability of Supervision

Supervised multidimensional projection methods take into account class label information to perform the mapping, aiming to place instances belonging to a same class close to each other in the visual space.

Typical methods such as Linear Discriminant Analysis (LDA) [50] perform the mapping by minimizing within-class distances while simultaneously maximizing between-class distances. LDA suffers from singular configurations that prevent the mapping to be properly performed, an issued that can be addressed with the use of regularization terms [61]. There are also alternatives that deal with singularities while producing mappings with uncorrelated attributes (similarly to PCA) [175].

Several MDP methods can be adapted to operate in a supervised manner. In fact, supervised variants of NeRV (a variant of SNE) [162], t-SNE [77] (another important variant of SNE), MVU [147], Isomap [53], LLE [178], and LSP [33] have been proposed in the literature. Most of those methods modify distances according to the class information, making instances from the same class closer while pushing apart instances in distinct classes. Therefore, MDP methods able to handle dissimilarity information can, at least in theory, be adapted to operate as a supervised method by first modifying dissimilarities according to class information. Another alternative is to employ a metric learning mechanism to estimate dissimilarities from labeled data before performing the projection [72]. Yet another approach is to leverage on the mapping distortions using two different techniques, one between classes and the other within classes [90].

Techniques such as LAMP [73] and Weighted-MDS [29], which make use of weighting coefficients to control the relevance of certain instances/distances in the mapping, can also be adapted to operate as a supervised method by just tunning weights according to class information.

2.3.4 Single- vs Multi-Level

Hierarchical mechanisms have been one of the alternatives used to play with large datasets, allowing to lessen the computational burden and the issue of generating cluttered layouts. Most hierarchical MDP (HMDP) techniques operate in two steps, first building a hierarchical representation of the data and then mapping data from particular levels of the hierarchy to the visual space. Since hierarchical projection methods rely on conventional MDP technique to perform the mapping from the hierarchy, we consider HMDP as variants of MDP techniques, pointing the existing variants by a • in the column "Multi-level" in Table 1.

Not every technique can be adapted to handle hierarchical data. The main issue is how to "synchronize" the mapping from different levels so as to preserve context and neighborhood structures. The synchronization mechanism highly depends on the properties of the MDP method used to perform the mapping. For instance, HIPP [116] relies on hierarchical cluster to organize the data, using cluster representatives from coarser levels as references to position representatives in finer levels, ensuring visual context and neighborhood preservation. HIPP builds upon control points supported by LSP [117] to ensure such synchronization. Therefore, any technique supporting control points can be used to enable the hierarchical projection scheme. Glimmer uses a GPU accelerated version of Chalmer's [32] method to perform the mapping, using decimation and interpolation schemes to build the hierarchy. A similar construction can be done using the force-direct mechanism [152]. Pezzoti et al. [121], in contrast, builds the hierarchy based on random walks on the neighborhood graph of the data, using t-SNE [157] as projection scheme.

2.3.5 Local vs Global

The concept of locality is used in the literature to characterize two different properties of MDP methods, namely *local modeling* and *local mapping*. The former, local modeling, stands for methods that aim to first extract local geometrical and/or topological information from the data, computing a global mapping that preserves such local information in a second step of the process. In other words, local modeling methods rely on neighborhood information of each data instance to build a single map that project the data while preserving the obtained neighborhood information. LLE [129], SNE and its variants [158], LSP [117], and Laplacian Eigenmaps [15] are examples of methods that relies on local modeling to yield global transformation of data.

Local mapping techniques also rely on neighborhood information but, in contrast to local modeling methods, they build a family of local transformations to project the data, that is, instances are not projected by a single global transformation but by a set of local mappings. Each mapping is responsible to project a subset of the data, preserving the corresponding local structures. Examples of local mapping methods are PLP [115] (a variant of LSP [117]) and LAMP [73]. In fact, LAMP pushes the concept of locality to the limit, using a different affine transformation to map each instance of data.

2.3.6 Steerability

Steerability accounts for the property of driving the projection process based on particular subsets of instances. Weighting mechanisms has long been a resource used to enable steerability [23], allowing that certain instances have larger influence in the mapping process. Tunning weights to steer projections is not an easy task though. Most algorithms find appropriate weights by first (interactively) placing a subset of instances on the visual space, from which a stress function is defined and minimized to find proper weights [66, 67]. The minimization scheme can easily become computationally costly, impairing interactive applications.

First introduced by LSP [117], the so-called control points are another resource employed to steer projections according to user interaction. Control points can be seen as user-specified "anchors" whose position in the visual space dictates the behavior of the projection process. The concept of control points does not rely on costly optimization procedures, thus making interactive layout updates feasible. The concept of control points has further been exploited by variants

of LSP [33, 115] and several other MDP methods. In particular, the LAMP [73] technique enables a high degree of steerability by combining control points with local mapping. Another characteristic of LAMP is that a reduced number of control points is needed to properly steer the projection [114]. However, choosing an appropriate set of control points so as to generate layouts with low distortion is a difficult problem that has barely been tackled in the literature [2], deserving further investigation.

Methods such as Pivot-MDS [24] and Chalmers [32] rely on the concept of "pivots" (or landmarks). Although pivots have originally been proposed as a speedup resource, they can also be adapted to incorporate some degree of steerability in the projection [70].

A recent variant of SNE, called A-tSNE [122], allows users to steer the accuracy of the mapping process based on regions of interest. However, in contrast to other steerable methods, users are not allowed to interactively change the layout by moving points around or changing the influence of particular instances in the projection.

2.3.7 Stability

Stability is a term that encompasses many different meanings, from sensitivity to data perturbation to guarantee of convergence to a global optimum. In the following discussion, stability refers to how responsive to variations in the input data a MDP method is, where the word "variations" accounts for any change in the data, since small perturbations to an increase/decrease in the number of instances to be mapped. If the mapping is stable then small variations in the data should lead to small variations in the projection. Furthermore, a stable method should not modify the position of projected instances when new data is handled.

Techniques that rely on eigen-decomposition, such as PCA, Isomap, and LLE, are not stable as to data variations. The reason is that the addition/removal of a few instances can significantly change eigenvectors and eigenvalues, thus affecting considerably the mapping (see [29] for a detailed discussion). Even small perturbations in the data can greatly impact in the mapping due to the flip of eigenvectors (u and $-u$ generates the same eigenspace), resulting in quite different layouts before and after the perturbation. It is important to make clear that when we say "addition/removal" of new data we are not talking about out-of-core data (discussed in the following subsection), but considering the case where the projection of a data set is compared against the projection of the same data set augmented (or reduced) with additional data items.

Techniques such as SNE, Kruskal, and Force-directed are based on optimization procedures whose optimal solution depends on initial conditions. Typically, different initialization take to different solutions. Stability is not granted even by fixing the initial condition, as the energy function to be minimized usually depends on the number of data instances, so a change in the number items may lead to substantially different layouts [49].

Techniques that rely on landmarks or control points, such as LAMP [73] and L-MDS [144] tend to be stable as long as control points and landmarks are kept fixed. Since the position of each instance only depends on the control points (landmarks), if those points do not change, the mapping also does not change. LSP [117] and its variants [33, 115], and Chalmers [32] are control point (landmark) based methods whose stability is more vulnerable. LSP is stable as to small perturbation on the data but it can result in completely different mappings if the number of instances increase or decrease drastically. The reason is that LSP performs the mapping based on neighborhood relations and such relations can change due to addition (removal) of new data items. Chalmers can be made stable if the initial condition is made fixed.

2.3.8 Out-of-Core Data

Capability to handle out-of-core data means the MDP technique is able to map new data preserving their relation with previously project data, that is, the MDP method can operate in a streaming fashion. The most straightforward example are linear techniques, which can naturally project new data using the same linear transformation computed to project previous data. Techniques such as NLM, Smacof, and SNE, which are based on iterative optimization procedures, can be adapted to handle OOC data by simply updating the cost function to take into account the variables associated to the new data. Techniques based on spectral decomposition such as Classical MDS and LLE are harder to adapt to the context of out-of-core data, since the eigen-decomposition is performed over the whole set of data [16]. However, pivot/landmark versions of those spectral methods, such as L-MDS [144] and Pivot-MDS [24], can naturally handle OOC data. Local mapping techniques such as PLP and Lamp can also handle OOC data in a very straightforward manner.

The main issue when using or adapting previously computed transformations to map OOC data is that bad quality mappings might be obtained if the new data presents very different properties when compared with the data that gave rise to the transformation. For instance, when using techniques that rely on control points, landmarks, and pivots, one cannot be sure if the chosen "driving" points are good representatives for the new data. An alternative is to recompute the whole projection when the quality of the layout reaches unacceptable levels, which can be measured and visualized on-the-fly [4, 6] (see Section 5.2 for more details).

2.3.9 Computational burden

Assuming the visual space is two or three-dimensional and that the number of instances n is greater than the number of dimensions p, the right most column in Table 1 summarizes the computational complexity of MDP techniques, where r stands for the number of control points (or landmarks) and i for the number of iterations (for iterative techniques). SOM, GTM, and CCA also depend on the number l of prototype points. GTM requires the inversion of an $m \times m$ matrix, where m is the number of basis functions. As one can easily see from Table 1, most MDP techniques are of quadratic and cubic order, rendering them not attractive for interactive applications. In fact, some techniques turn out prohibitive even when handling data sets with only a few thousand instances.

Different mechanisms have been proposed to speedup MDP techniques, being the used of landmarks [144] and pivots [24] a common alternative. Methods that rely on landmarks and pivots compute a mapping considering the (dis)similarity between instances and landmarks/pivots, reducing the computational burden from n^2 (or n^3) to rn (or nr^2), where the number of landmarks r is typically much smaller than n. Numerical algorithms devoted to exploit sparseness can also be employed to speedup eigen solvers and linear system solvers [111], bringing the complexity of methods such as LLE, Lapl. Eigenmaps, and LSP close to $O(n^2)$, as those MDP methods operate on sparse matrices.

GPU-accelerated versions of MDP techniques have also been proposed, as for example Glimmer [70], a GPU-based version of Chalmers' method, and CFMDS [113], a GPU implementation of Classical MDS. Sampling combined with progressive updates [171] and with interpolation [101] are also alternatives employed to expedite MDS methods.

As previously discussed, MDS methods based on nonlinear optimization tend to be computationally costly. Different strategies such as vector extrapolation [127], quick gradient computation [69], and spacial data structures to fast neighborhood approximation [157] have been used to speedup the optimization process.

Many of the MDP methods discussed in this section have their code available for download. A list of available codes, their programming languages and corresponding URLs is described in the supplementary material accompanying this manuscript.

2.4 Which MDP to use given Visualization Factors?

In the visualization literature, most authors justify the choice of a particular MDP technique based on unique criteria such as "computational efficiency", "layout steerability", or "linearity", missing a deeper discussion on how the specificities of the problem under analysis can impact in the choice of a particular MDP method. One of the reasons for such slight approach is the lack of a systematic procedure to support users in their choice of an appropriate MDP technique. The proposed taxonomy is a first step towards filling this gap, enabling users to better

make out with MDP methods. More specifically, the proposed taxonomy allows for an incremental approach where users can start with a few intrinsic properties (constraints) of the problem under investigation, progressively adding new traits that the MDP method should comply with.

For instance, as attribute type has an impact on visual channel (Bertin typology), **data type** impacts on similarity measures and thus the resulting layout. Assuming the application deals with Cartesian data, users can opt to one of the 48 MDP methods able to handle Cartesian data (the numbers mentioned in this section reflects only techniques discussed in this survey). However, they can also transform Cartesian data to dissimilarity data, with the additional flexibility of designing transformations to "distort" the Euclidean metric in certain items or regions. In the latter case, users can decide among the 36 candidates able to deal with dissimilarities. As we can see, except for ordinal and categorical data, only the data type information is not enough to justify the choice of a particular MDP technique.

If the visual analysis of **linear trends** and/or the **relation between axes and data attributes** are relevant for the application, linear projections become the natural choice. In this case, converting Cartesian data to distances does not make much sense, so the sought techniques should be among the linear ones able to handle pure Cartesian data. In this scenario users have only 5 alternatives: PCA, iPCA, PCP, PLMP and Kelp, assuming that PLMP and Kelp have been combined with a linear method to place control points and that a linear kernel is being used in Kelp.

When data is endowed with **labels**, one expects visualizations where class structures are well defined and class-outliers are clearly seen. Visualizing if classes are separated or overlapped, or whether particular classes are split in several components in the data space, is also important. Assuming the data can be of any type and that methods can either be linear or non-linear, the proposed taxonomy points to 7 supervised MDP that can natively play with labeled data: LDA, ClassiMap, SLLE, S-Isomap, MUHSIC, supervised extension of NeRV, and DS-tSNE. However, within-class structure tends to be better preserved by techniques bearing the locality property (local modeling techniques), remaining only two possibilities: SLLE and MUHSIC.

Locality is in fact a desirable property in many applications, mainly when combined with **steerability**. The reason is that local edits accomplished by users in a projection layout typically affects the position of several points, which have to be remapped. Local techniques tend to keep static parts of the layout not affected by user intervention, thus preserving context and user's mental map. The combination "locality + steerability" is also useful for exploratory tasks, allowing users to perceive cause and effect and so better understand what a MDP method is accomplishing, how sensitive it is to small changes in the display, and how points are "connected" to each other. This fact calls for "common fate", that is, things that behave similarly are instantaneously associated with low cognitive effort. Therefore, looking in our taxonomy for steerable and local methods we find 8 techniques: LSP, PLMP, LAMP, LoCH, Kelp, PLP, E-LSP, and HIPP. If we add Multi-level as an additional constraint, which is fundamental to incorporate the well-known Schneiderman's mantra "overview first, zoom and filter, then details-on-demand" [140] into the context of multidimensional projection, we end up with a single MDP technique, namely, the HIPP method. Other techniques could be adapted to operate in multi-level, but HIPP is the only one that natively matches the properties of being local, steerable and multi-level.

Stability is important for several reasons, as for example, ensuring trust, reproducibility of results and removing the uncertainty due to the MDP process. It also allows to preserve context and mental map, as neighbor points on one view are still close on another when different parameters are used, reinforcing the perception of which points are actually "similar". In particular, by combining stability with locality and steerability, one can build a reliable MDP-based system for high-dimensional data analysis. The provided taxonomy shows that only LAMP matches stability, locality, and steerability requirements simultaneously, although other techniques such as LSP and E-LSP can ensure them if the insertion and removal of data instances are not allowed.

Computational times is essential for interactivity. Visual analytics is about working memory, we think about a hypothesis and want to see whether it is true. The search for an answer might involve the exploration of parameters and interaction with the resulting visualization. If the MDP takes too long to render the layout, users might forget what they are expecting, hampering their mental model. If steerability is needed, computational times are even more crucial, as users expect real time updates during interaction with control points. Computational time is a complex issue in the choice of a proper MDP method, as effective techniques such as BH-tSNE and LAMP can slow down quickly depending on the number of points and data dimension. As a guideline, we recommend users to select a set of candidate techniques using our taxonomy, balancing the impact of computational time as the last step in the choice process.

3 Distortions and Metrics

Since isometric mappings from high-dimensional to lower dimensional spaces are only possible under very particular conditions, errors and distortions are highly prone to be present in MDP layouts. In other words, neighborhood structures observed in the visual space might not be exactly the same ones existing in the original multidimensional space. Considering that MDP-based visual analysis assumes that proximity relations in the visual space reflect similarities and that our visual perception is biased toward the validity of this assumption (Gestalt law of proximity), the possibility of distortions between original and visual neighborhoods introduces uncertainties that impact the human analytic process [131]. Characterizing and quantifying errors and distortions introduced by MDP methods is thus a major concern in the context of visualization.

In this section we provide a discussion on how distortions can impact different stages of MDP-based visual analytic processes, describing the different types of distortions that might be present in a MDP layout as well as existing metrics for quantifying them.

3.1 MDP distortions as uncertainty in the visual analytic pipeline

Referring to the *Knowledge generation model for visual anaytics* [130] and the uncertainty analysis in this model [131], MDP can be seen as a modeling process Φ that transforms the original data $\{\mathbf{x}_i\}$ or $\{\delta_i\}$ into points $\{\mathbf{y}_i\}$ with the *Model* component (the Model stage), mapping those points to graphical variables to form the scatter plot idiom in the *Visualization* component (the Visualization stage). The uncertainties associated to both stages of this visual analytic pipeline correspond to distortions generated by the MDP, which we shall analyze in more depth now.

3.1.1 MDP distortions at the Visualization stage

At the visualization stage, data items are encoded as points positioned by the MDP in the visual space. Beyond typical uncertainties caused by clutter, resolution, and contrast in scatter plots, an uncertainty source specific to MDP is that the effectiveness of a MDP analysis relies on the Gestalt law of proximity, so on the pre-attentive perception of the relative position of the points rather than on their absolute position within a frame of reference.

For this reason, regarding the cognitive side of the visual analytic process, and for the sake of the *expressiveness principle* [103] which requires that all of, and only, the information in the data should be expressed by the visual encoding, it is advisable to remove any frame of reference like the orthogonal axes from the layout. Indeed, these graphical elements would perturb the analysis by forcing users to question the meaning of these unnecessary axes, leading them to read the absolute position of points in that frame of reference rather than focusing on their relative positions. The only case where keeping axes makes sense is when MDP provides a linear mapping Φ from the Cartesian data space, so axes bear some meaning relative to the original dimensions, as is the case of PCA.

Moreover, on the perception side of the visual analytic process, although there is no empirical evidence regarding scatter plots specifically produced by MDP, a circular frame containing the points and centered

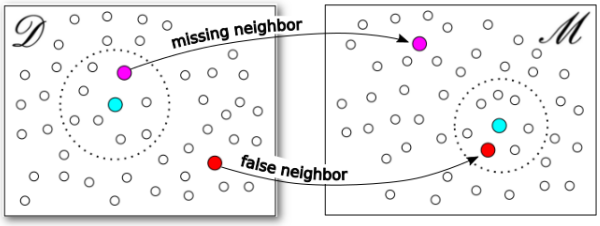


Fig. 1: Illustration of Missing and False Neighbor phenomena causing distortions and uncertainty of visual analysis based on MDP (Sec. 3.2).

on their center of mass as used by Kubovy [80] and discussed among other heuristics to select a frame of reference by Palmer [109], could be used instead of a square frame. It would avoid the known perceptual salience of the axes directions and symmetries [110] of a square frame, while still emulating its unit aspect ratio. This circular frame would also support the fact that similarities and point patterns have no *a priori* preferential directions, which is consistent with the original similarity data transformed by the MDP model.

Another important point is that the unit vectors of hidden horizontal and vertical axes used to draw the points shall be of the same length on the screen, so a circle drawn in that visual space should be rendered as a circle on the display. The reason is that in the ideal case of perfect matching between data similarities δ_{ij} and points proximities d_{ij}, and assuming isotropic pairwise distance perception by the user, for two points i and j perceived at some distance d_{ij} on the screen, the corresponding similarity δ_{ij} in the data space should be the same whatever the direction of the vector $(\mathbf{y}_j - \mathbf{y}_i)$.

Discussion above provides mostly heuristic guidelines that would still strongly benefit from visual perception studies to validate, invalidate or specify them.

3.1.2 MDP distortions at the Model stage

At the model stage, distortions occur for several reasons. As already discussed, it is nearly impossible for MDP to map the data as a set of points in the visual space (a plane) such that Euclidean distances are perfectly preserved. The first reason is data dissimilarities might be inherently *non Euclidean* (like ranks, probabilities, etc.) so the Euclidean visual space cannot express all the data variability due to metric constraints. Secondly, data may well be Euclidean but do not lie in a plane embedded in the multidimensional Euclidean space, so possibly nearby some manifolds with *intrinsic dimension* greater than 2, with *non-linearities* or *non trivial topology* (sphere, torus, branching structures). In that case some "squeezing" and/or "stretching" of distances during the unfolding of the nonlinear structures should take place. The third reason refers to the *curse of dimensionality* which shifts distance distributions in high dimensional space, so items looks all dissimilar to each-other, a phenomenon not observed in the visual space [85].

The distortions at the model stage can thus be observed in the fact that the layout distances D only approximate original dissimilarities Δ. These distortions can be, and shall be quantified to let the user takes informed decision when attempting to achieve analytic tasks. For instance, when inferring that some patterns (clusters, outliers, trends,...) exist in the data space determined by Δ, from their visual perception in the layout determined by D.

In the sequel we focus on these model-based MDP distortions and present metrics that have been proposed to quantify them.

3.2 Error Types

Distortions in MDP mappings are basically caused by two different phenomena that affect neighborhood structures in the visual space. The first phenomenon, which we call *Missing Neighbor* (MN), happens when neighbor instances in the original space $\mathscr{D}$ are mapped far apart each other in the visual space $\mathscr{M}$. The second phenomenon, called *False Neighbor* (FN), occurs in the other way around, that is, one or more items that are not neighbors in original space are mapped close to each other in the visual space. In the following, items that give rise to MN and FN phenomena will be called MN and FN items, respectively, while *True Neighbor* (TN) items are those not associated to MN and FN phenomena, that is, TN items are neighbors in both the original and visual spaces. The ideal MDP layout would have only such TN items. Figure 1 illustrates the concept of Missing and False Neighbors.

A natural question arises regarding how likely each MDP technique is to generate MN or FN phenomena. Unfortunately there is no clear answer for all MDP techniques. Still in a formal analysis conducted by Lee and Verleysen [85] [86], *plasticity* which favors MN and *elasticity* which favors FN, are key factors identified within the energy function of a family of MDP techniques (Sammon [132], CCA [43], SNE [64], t-SNE [158], NeRV [162]). ClassiMap [90] is a supervised MDP technique which takes advantage of these factors to generate MDP of labeled data by directing FN and MN distortions within and between classes respectively.

Overall, MN and FN phenomena result in deceptive neighborhood structures and, since such distortions are highly likely to take place, uncertainties are always present in the visual analysis process. The degree of uncertainty is directly related to the intensity with which MN and FN phenomena take place. Therefore, measuring the intensity of MN and FN phenomena is fundamental to gauge the uncertainty present in MDP layouts. The larger the distortion the more uncertain one can be about what is being analyzed.

A multitude of quantitative methods have been proposed to assess the intensity of distortions in MDP mappings. Most of those methods and their characteristics are discussed in the following.

3.3 Error Metrics

Quantitative measures to evaluate MDP distortions vary greatly as to the manner they operate. The different measurement methods aim to quantify distinct aspects of the MN and FN phenomena, demanding local and/or global inspections as well as mechanisms able to gauge particular traits of such distortions.

In order to properly characterize the methods devoted to assess MDP distortion, we propose two different taxonomies, *Span* and *Distortion Type*. Table 3 summarizes the quality measures considered in this survey and their properties according to the two taxonomies. Table 3 also shows the *Output* ranges and the optimal value for each measure, *i.e.* the value that indicates no distortion at all in a MDP mapping.

It is important to make clear that the quality measures in Table 3 are the ones that appear more frequently in the context of visualization. Other quality measures have been used [59], mainly to operate with supervised MDP techniques (labeled data) [102, 165]. Those methods, though, are not designed to evaluate MN and FN distortions, motivating us to not include them in Table 3.

3.3.1 Span

The *Span* taxonomy discriminates techniques that operate locally (local measures), i.e., in the neighborhood of each instance, from those that make use of global information to gauge the quality of a MDP mapping (global measures).

Evaluation schemes that rely on loss functions (Stress and Strain [29]) are typical examples of global measures, where the quality of a mapping is assessed by pairwise comparing (dis)similarity in the original and visual spaces. A main issue with loss functions is that those measures are not scale invariant, requiring a thorough control of scales in the visual domain to ensure fair comparisons between mappings. Topographic Product [14] is another scale dependent global method that relies on dissimilarities to assess MDP layouts. In contrast to loss function based schemes, which build upon pairwise differences, Topographic Product computes the product of dissimilarity ratios. Such ratios are computed element-wise according to the order of instances in each neighborhood, aggregating all the obtained values in a single final score. Loss functions and Topographic Product are mostly employed as a mechanism to compare MDP methods (for a fixed dataset), since their output, as a standalone metric, is not so easy to interpret. For instance, a Stress value of 0.5 does not mean much, since this value is scale sensitive. For a given data set, the only claim one can make

Table 3: MDP distortion measures and taxonomies described in Section 3.3.

Quality Measure	Span		Distortion Type							Output	
	Local	Global	Dissimil.	Corr.	Prob.	Rank	Geom.	Set Diff.	Homol.	Range	Best
Stress/Strain [29, 79, 132]		✔	✔							$\mathbb{R}^+$	0
Spearman's Correlation [141]		✔		✔		✔				$[-1, 1]$	1
Topographic Product [14]	•	✔	✔							$\mathbb{R}$	0
Correlation Coefficient [53]		✔	✔	✔						$[-1, 1]$	1
Trustworthiness/Continuity [161]	✔					✔				$[0.5, 1]^*$	1
Mean Rel. Rank Errors [83]	✔					✔				$[0, 1]$	0
LC Meta-criterion [34]	✔							✔		$[0, 1]$	1
Procrustes Measure [55]	✔						✔			$\mathbb{R}^+$	0
KL Diverg. [64, 162]	✔				✔					$\mathbb{R}^+$	0
Global Quality Q_y [99]	✔	✔	•	✔		✔	•	•		$[-1, 1]$	1
NIEQA [177]	✔	✔					✔			$\mathbb{R}^+$	0
Graph-based Family [102]	✔	•	✔					✔		$[-1, 1]$	1
Smooth. Neigh. Pres. [108]	✔		✔			✔				$[0, 1]$	0
Persit. Homol. [125]		✔							✔	$\mathbb{R}^+$	0

*: the lower bound of 0.5 is conjectured.

is that a mapping with Stress 0.5 is globally better than a mapping with Stress 1.0, assuming that both layouts are in the same scale. A mechanism commonly used to avoid the scaling issue is to force the visual domain to be a unitary square. However, such "normalization" can also drastically affects the values resulting from quality metrics, mainly when outliers (instances mapped far apart from other instances) are present in the layout.

The Spearman's correlation [141] and Correlation Coefficient [53] are global distortion measures that rely on Pearson's correlation to gauge the quality of a mapping, making them scale invariant. Moreover, the output of those methods is easier to interpret, since values close to 1 indicate a good correlation between dissimilarities in the original an visual spaces, giving some estimate (although not precise) of the uncertainty present in the layout.

The homology-based approach proposed by Rieck and Leitte [125] uses a Rips graph in both data and visual spaces, computing their persistent homology which encodes the number of connected components at different scales. The distance between the persistence diagrams is used to globally quantify the mapping distortions.

Local quality metrics assess distortions by comparing neighborhoods in the original space against their counterpart in the visual space. Those techniques typically operate element-wise, measuring the intensity of MN and FN phenomena in each neighborhood, aggregating the computed distortions in a single quality measure that evaluates the MDP layout as a whole. A typical example is the graph-based approach proposed by Motta et al. [102].

There are also methods such as Q_y [99] and NIEQA [177] that combine both local and global distortion analysis. Those techniques make use of neighborhood information to construct a nearest neighbor graph from which distances are globally compared through correlation analysis. Therefore, those methods are able to gauge local as well as global distortions simultaneously.

3.3.2 Distortion Type

The *Distortion Type* taxonomy accounts for the type of distortion measured by each quality metric. The distortion type *Dissimilarity* indicates the metric accounts for how the value of dissimilarity between instances are affected by the mapping. *Rank* stands for the order of elements, that is, how much the order of elements changes due to the mapping. Metrics focused on *Correlation* assess how correlated some property (dissimilarity, rank, etc.) in the original space is to its counterpart in the visual space. *Probability* distortion type accounts for distortions in the distribution of probability of some variable (dissimilarity, for instance) after the mapping. Methods devoted to measure *Geometric* distortions typically gauge how neighborhoods are stretched/shrunk by MDP. *Set Difference* simply compares, for each instance, how many of its neighbors remain neighbors in the visual space. Finally, *Homology* compares persistence diagrams that encode the connectedness of the data across different scales in both spaces.

Stress/Strain, Correlation Coefficient, and Topographic Product, measure distortions as to dissimilarities, although using distinct mechanisms. Stress/Strain assesses distortions by computing pairwise differences between dissimilarities. Topographic Product relies on element-wise product of ratios between dissimilarities (if only the nearest neighbors are considered for the ratios the method becomes local). Correlation Coefficient computes the Pearson's correlation between pairwise dissimilarities in the original and visual space. Spearman's correlation [141] also measure distortion as to dissimilarity, but not directly, that is, dissimilarities from the original and visual spaces are independently sorted and the ranks in the sorted lists are used to compute the correlation. Therefore, Spearman's correlation is measuring how correlated the ranks of the dissimilarities are, not the dissimilarities themselves.

Table 3 also reveals that global techniques mainly assess how much dissimilarities or their correlation are affected by the dimensionality reduction process (notice the simultaneous check marks Global+Dissimil. and Global+Corr.). Local techniques, in contrast, have a wider scope as to the type of distortion they are able to evaluate. Trustworthiness/Continuity [161], for instance, measures how the rank of instances in each neighborhood is affected by MN and FN phenomena. Therefore, Trustworthiness/Continuity assesses the quality of a layout based on local changes, assessing directly the intensity of MN and FN phenomena. Mean Relative Rank Errors [84] and LC Meta-criterion [34] are closely related to Trustworthiness/Continuity in the sense they measure distortions based on neighborhood information, but focusing in different facets. Mean Relative Rank Errors quantifies distortions by analyzing how much the rank of elements in each neighborhood changes, but considering the whole set of neighbors, not only MN and FN instances as done by Trustworthiness/Continuity. LC Meta-criterion simply measures the percentage of True Neighbors in each neighborhood, without taking into account the rank of the elements. Trustworthiness/Continuity, Mean Relative Rank Errors, and LC Meta-criterion are all variants of the so-called precision-recall measures [162] employed in the context of information retrieval. Moreover, as shown by Lee and Verleysen [84], those three measures and other variants [100] can be derived from a single structure called the Co-Rank matrix.

Procrustes Measure [55] assumes the original data is embedded in a Cartesian space. The neighborhood of each instance is first conformally mapped to a two-dimensional affine subspace and the mapped neighborhood is then matched with its counterpart in the visual space via rigid motion transformation. The difference between them is used to gauge the distortion introduced by the MDP mapping. Similar to Stress/Strain, Procrustes Measure is not scale invariant, thus being more appropriate to compare different MDP methods than for the individual analysis of a single mapping.

Quality measurements based on probability distortions rely on KL Divergence [64] to evaluate MDP. Venna et al. [162] showed that KL-divergence can be interpreted as probabilistic versions of precision-recall measures, thus also making a parallel between MDP layouts assessment and information retrieval evaluation.

As mentioned before, the Q_y [99] method enables a flexible mechanism to combine local and global distortion analysis. Although the original version of Q_y relies on Spearman's Coefficient to quantify local distortions, other methods could be used for the same purpose, as indicated by • in Table 3.

Motta et al. [102] derived a set of metrics from a graph-based framework. The graph-based metrics are indeed variants of methods such as LC Meta-criterion [34] and Mean Relative Rank Errors [84] that operate on a graph-structure called Extended Minimum Spanning Tree.

Techniques such as Trustworthiness/Continuity, Mean Relative Rank Errors, and LC Meta-criterion operate in a "discrete" manner, i.e., they use discrete information such as the number of matches or the rank-order of the elements to measure how well neighborhoods are preserved during the mapping. A smooth variant of Trustworthiness/Continuity, called Smooth Neighborhood Preservation, has been proposed by Pagliosa et al. [108]. The reasoning is to discriminate MN and FN that are mapped close to their true neighborhood from those mapped far apart, weighting the error accordingly.

Other methods. Visual mechanisms such as Shephard's diagrams [138] have been used to qualitatively evaluate MDP mappings. Shephard's diagram turns out useful when the methods under comparison differ considerably in terms of quality. When data is endowed with class information, methods originally designed to evaluate the quality of clusters have been adapted to assess MDP mappings [73, 96, 145]. Those methods, though, are not designed to evaluate MN and FN distortions, but how much a MDP mapping preserve the class structures in the visual space.

3.4 How MDP distortions may impact the visual analysis

We saw that a substantial amount of work has been done towards proposing metrics that quantify distortions in MDP mappings. However, very few is known about how those distortions impact the MDP-based analytic tasks.

One of the few works in this context is the experiment accomplished by Lewis et al. [91], where a user study was conducted to find out whether expert data analysts, informed novices, and novices, are consistent in evaluating the "quality" of a map. They were all given *no specific analytic task* except the experimenters pretended they will have one to conduct. They also had been given on purpose no precise definition of what "quality" means. Experts and informed novices were made aware about the data characteristics, and only experts knew about the MDP techniques. As a main result, MDP experts were found consistent judges positively correlated with the trustworthiness mapping quality measure whereas other users were not. As many MDP users are not MDP experts there are still work to be done in training them, or more practically, to guide them in interpreting MDP.

Regarding PCA, it is well known that it preserves global structure more than local ones [29], as it is designed to preserve large variance hiding small ones that occur orthogonally to the two principal axes. That means small scale visual patterns are more uncertain than large scale ones. But the impact of the proportion of variance explained by the first two principal components on different analytic tasks has not been explored and quantified. Another important discovery has been the shift-invariant similarities [85] that t-SNE and other similar techniques exploit, making them more robust against norm concentration. Norm concentration is a typical characteristic of the *curse of dimensionality*: in high dimensional spaces, the variance of all pairwise data dissimilarities decreases toward 0 while their average value still increases when the dimension p increases. This means that data look all dissimilar to each others. This leads MDP of high-dimensional data to produce maps with no salient pattern, where all points concentrate in the center of the map generating clutter and possibly hiding local structures. The use of shift-invariant similarities limits this effect for theoretically grounded reasons [85], making patterns in the data more likely visible in the map.

In the following section we shall further examine the interplay between MDP distortions and the kind of analytic tasks performed in MDP layouts. This analysis will allow us to better understand how MDP distortions may impact these tasks, at least qualitatively.

4 Analytic Tasks

As world map projections [156] have been designed to preserve angles (Mercator conformal projection) for easier navigation, or to preserve areas for political fairness (see [160] for more exotic maps), it is important to consider the intended analytic tasks to decide about the best MDP. In this section, we propose a taxonomy of analytic tasks involving MDP scatter plots. We focus this taxonomy on a formal definition of the tasks that allows us a qualitative analysis to identify which of them can possibly be impacted by MDP distortions. This is crucial as some intended tasks *cannot* properly be achieved by the user from the base MDP layout, while many non-expert users are unaware of this fact. We aim to raise user awareness about MDP uncertainties, identify these specific ambiguous tasks and their non-ambiguous counterpart.

Thanks to that task taxonomy and its formal description, we can also further match the analytic tasks with the MDP taxonomy described in Section 2.3, thus providing guidelines to choose an MDP given a task.

Beyond setting a taxonomy, our formalization of the tasks starts to pave a path towards quantifying, possibly via user studies, the impact of distortions in typical analytic tasks involving MDP layouts, so as to enable systematic evaluation of the efficiency of using a specific MDP for a given task.

4.1 Taxonomy of MDP Analytic Tasks

In the following, we review the literature surveyed in the recent work of Sacha et al. [131], but we focus on identifying the analytic tasks involving MDP layouts. Specifically, the study by Sacha et al. has a broader scope than ours, analyzing the full Visual Analytic interaction loop which enables the user to visualize the map and interactively change the parameters of the whole visualization pipeline, including the data and the MDP model. Here we focus on analytic tasks based on a single MDP layout, so we ignore tasks involving multiple coordinated MDP views. Our taxonomy extends and refines the one based on interviews of data scientists proposed by Brehmer et al. [26]. Our main goal is to determine qualitatively which analytic tasks can or cannot be hampered by MDP distortions, and further provide a matching with MDP taxonomy to recommend MDP techniques which would best fit with the intended analytic task.

We ought to get a *formal description* of the tasks that allow designers and users to clearly distinguish between feasible and infeasible tasks achievements regarding a target pattern. For instance, we want to make clear that we cannot expect a user to identify *with certainty* an outlier in the data by locating an outlier in the base MDP layout, because the map is a distorted representation of the data. However, we can provide users with the *necessary* information to identify an outlier in the data by showing a map *enriched with original data similarity* information. This kind of analysis has never been proposed so far.

For this sake, we use the Munzner's `What/Why/How` typology of visualization [25, 26, 103] to provide a non ambiguous task description, empowering this semi-categorical description with a mathematical formalization of the input and output spaces of the tasks and the entities they transform. An attempt for such a formalization has been initiated in the "Pro Format Abstract" taxonomy of analytic tasks [1], but not with enough details to achieve our goal, and for tasks that are not specific to MDP layouts. The extensive description of the tasks in terms of Munzner's typology is provided in supplementary material.

To determine the taxons of our taxonomy, we analyzed all the papers from the literature survey of Sacha et al. [131] and determined common tasks and subtasks progressively. We attempted for each paper to develop a pseudo code as if a computer should realize the same task as described in the paper, using the Munzner's typology as much as we could as coding instructions. We had in particular to identify unambiguously the input and output elements and spaces of the tasks, and use mathematical formalization for this sake. It helped us to detect similar or identical tasks to be merged and progressively converge to a set of unique canonical tasks reported in Table 4.

As described in details in the Supplementary Material, some tasks are composed of sub-tasks or used in sequence in the literature. We connect these tasks to each other using numbers nearby task names in Table 4. In that table, the `Input` and `Output` columns refer to the space or set of

elements used as input to the task and output it provides. For instance, in the *Discover Clusters in Map* task, the user must consider as input the n projected points in $\mathscr{M}^n$, then `Locate` and `Identify` clusters as output which consists formally in assigning the n points to one out of the k clusters $\mathscr{C}_k$. A set is underlined when the items are identified in the user's mind but neither recorded in the machine nor `Annotated` in the view *per se*. Along this line, the `Actor` type identifies whether the machine (M) or the user (U) accomplishes the task. These notation distinguishing between humans and machines are important to inform the global design of the visual analytic system: to identify whether a data item is *objective* (it can be shared between different processes) or *subjective* (it lies in the user's mind); and to determine whether a task shall be implemented with some computing technique evaluated with computer simulations (M), or requires some user interaction evaluated with user studies (U). The TS column identifies the main space into which a task operates ($\mathscr{M}$ or $\mathscr{D}$) and emphasizes the risk of misuse of MDP for certain tasks ($\mathscr{M}$!). More details are given in Section 4.2. The other columns refer to the taxons of the MDP taxonomy described in Sections 2.3 and 2.4 and can be used to match the intended task with the best suited MDP as discussed in Section 4.3 and in Table 4.

We provide a quantitative summary of the tasks with a count of the surveyed papers which use them in Table 5. The full list of papers associated to each task is provided in the supplementary material. In summary, we have identified four categories of tasks:

- **Generate Map**: Tasks of this category are accomplished by the machine to generate different MDP layouts;
- **Explore Dimensions (Axes)**: Tasks of this category apply to MDP layout data with available features $\mathbf{X} \in \mathscr{D}_x^n$ (typically data in Cartesian spaces) or external features $\underline{\mathscr{Z}}$ (typically external user knowledge not explicitly in $\mathscr{D}_x$);
- **Explore Items in Base Layout**: Tasks of this category apply to non-enriched MDP layout showing only the items' location with no additional information;
- **Explore Items in Enriched Layout**: Tasks of this category apply to enriched MDP layouts showing items and additional information encoded as colors, shape or text for instance.

In the sequel we first analyze which tasks can be impacted by distortions and which are immune to them. Then we provide guidelines to determine which MDP is best to use for an intended task.

4.2 Which Tasks Require Careful Map Interpretation?

The formal input and output description of the tasks summarized in Table 4 allows us to objectively identify the ones sensitive to MDP distortions and the manner they are so. We do not consider the *Generate Map* tasks (Table 4) in the following discussion, as they do not involve user (U) visual perception *per se*. Notice that these map-generating tasks are surveyed in Section 2 and are the ones onto which most of the Machine Learning literature about MDP focuses its efforts.

From Table 4 we observe two main cases summarized in the Target Space (TS) column regarding the data $\mathbf{X}$ or Δ ($\mathbf{X}|\Delta$), and mapped dissimilarities D :

- **Map-targeted tasks**, denoted as $\mathscr{M}$ or $\mathscr{M}$! in TS column. Only the map is given as `Input`, which means users have visual access to D but not to $\mathbf{X}|\Delta$.
- **Data-space-targeted tasks**, denoted as $\mathscr{D}$ in TS column. Data is provided together with the map $\mathscr{M}$ as `Input`, so users have visual access to all D plus part or all of $\mathbf{X}|\Delta$.

The **Map-targeted tasks**, whose *formal* description builds the `Output` without considering information from the data space $\mathscr{D}$ as `Input`, so cannot support inference about the original data $\mathbf{X}|\Delta$ but only about their MDP distorted, uncertain, representation D.

As discussed by Sacha et al. [131], the user can be aware of the MDP model uncertainties. In this case, s/he will use complementary trustworthy views like SPLOM, parallel coordinate plots, or will enrich the base layout with "thumbnail images" of the original data $\mathbf{X}|\Delta$ to get knowledge about whatever patterns (or non-patterns [10]) appear in the MDP layout. Thus, the *uncertainty-aware* user does not rely on the base MDP layout to reveal actual patterns in the data, but uses it as a data proxy that helps generate hypotheses regarding potential data patterns to be verified using the other trustworthy views.

Otherwise, *users unaware* of MDP model uncertainties about D as representing $\mathbf{X}|\Delta$, and furthermore, *mistaken users* who over-trust the map wrongly believing it is correct, are both prone to erroneous interpretations. They will take erroneously for granted that patterns (or absence of them) of points in the map $\mathscr{M}$ account for the same existing patterns (or absence of them) in the original data space $\mathscr{D}$.

Notice that the knowledge one can expect to get about $\mathbf{X}|\Delta$ by visualizing coordinated views from different MDP techniques, none of which being trustworthy, with such Map-targeted tasks only, is still uncertain. For instance, the user can annotate as $c_1 \in \mathscr{C}_1$ a cluster of points found with *Discover Clusters in Map* ($\mathscr{M}_1^n \to \underline{\mathscr{C}_1^n}$) in the first uncertain map $\mathscr{M}_1$; then s/he can evaluate positively with *Match Clusters and Classes in Map* ($(\mathscr{M}_2 \times \mathscr{C}_1 \times \underline{\mathscr{C}_2})^n \to \mathbb{R}^+$) that these class-color-coded items $\mathscr{C}_1$ form a single pure cluster $\underline{\mathscr{C}_2}$ in a map $\mathscr{M}_2$, while $\mathscr{M}_2$ is uncertain as well. And still, s/he has no way to be certain that this cluster $\mathscr{C}_1$ exists in $\mathscr{D}$, as $\mathbf{X}|\Delta$ never appears as input in any of these analytic tasks. Therefore, points in $\mathscr{C}_1$ could come from $K > 1$ distinct clusters in $\mathscr{D}$, overlapping in both $\mathscr{M}_1$ and $\mathscr{M}_2$. In short, composing several *uncertain* outcomes can hardly make the composite outcome more *certain*.

Therefore, for the *Map-targeted tasks*, the more distorted D is relative to $\mathbf{X}|\Delta$, the more error prone the task is. Table 4 shows, in particular, that a MDP is at high risk of misuse by users *over-trusting* the map when accomplishing the tasks denoted as $\mathscr{M}$! in TS column, because they are semantically *not space-specific*. Thus, users may erroneously think they are performing the task in the data space instead of actually in the map. We shall call them *Ambiguous Map-targeted tasks* ($\mathscr{M}$!-tasks) in the sequel. The list of these tasks is:

- Discover Neighbors (in Map)
- Brush (in Map)
- Navigate (in Map)
- Discover a Path (in Map)
- Discover Clusters (in Map)
- Discover Outlier (in Map)
- Find Class-Outlier (in Map)
- Evaluate Cluster Purity (in Map)
- Evaluate Class Compactness (in Map)
- Match Clusters and Classes (in Map)
- Classify Out-of-Core Item (in Map)

However, in the following cases, the fact the map only approximates the data cannot be ignored by users because of the very objective of the task as made clear by its name. The list of these *Non-ambiguous Map-targeted tasks* ($\mathscr{M}$-tasks) is:

- Name Synthesized Dimension
- Discover a Seed Point
- Steer Projection (and its subtasks)
- Sample Data Space from Map (and its subtasks)

Regarding the **Data-space-targeted tasks** ($\mathscr{D}$-tasks), the map layout is enriched with part or all of $\mathbf{X}|\Delta$ and other information required for the task. Compared to the map-targeted tasks, we know at least that $\mathbf{X}|\Delta$ ($\mathscr{D}$) is visualized together with D ($\mathscr{M}$). Therefore, some necessary information is displayed that theoretically allows users to visualize the task-targeted data directly to compare both map and data similarities. Thus, it enables at least a visual evaluation of local distortions.

Considering the Munzner's typology of tasks, the `What` refers to the data input necessary for the user to achieve the task. The above analysis

Table 4: Linking Analytic Tasks and Techniques (see complete caption on page 12).

	Intended Analytic Tasks	**Input space/set**	**Output space/set**	**Actor**	**TS**	**MDP property**	**Best matching MDP** (see section 4.3)
Generate Layout	Dimension Synthesis	$\mathscr{D}^n_{x/\delta}$	$\mathscr{M}^n$	M		$\star$	Any but (MCA)
	.1. Out-of-Core Extension	$\mathscr{D}_x \times (\mathscr{D}_x \times \mathscr{M})^n$	$\mathscr{M}$	M		OOC	PCA,LDA,SOM,CHLM,CCA,FDP,LAMP,LoCH
	Map Data with Intermediary Landmarks	$\mathscr{D}^n_x \times \mathbb{N}$	$\mathscr{M}^n$	M&U		Landmarks	{Landmark MDP}$\cup${L-MDS,Pivot-MDS}
	- Define Data-Landmarks Given Data	$\mathscr{D}^n_x \times \mathbb{N}$	$\mathscr{D}^r_x$	M&U		–	{Landmark MDP}={LSP, PLMP,
	- Map Data-Landmarks	$\mathscr{D}^r_x$	$\mathscr{M}^r$	M		–	LAMP, RBF-MP,LoCH, Kelp, PLP}
	- Map Data Given Data-Landmarks	$\mathscr{D}^n_x \times (\mathscr{D}_x \times \mathscr{M})^r$	$\mathscr{M}^n$	M		–	
	Map Labeled Data	$(\mathscr{D}_{x/\delta} \times \mathscr{L})^n$	$\mathscr{M}^n$	M		Supervision	{Supervised MDP}
	Map Items Relative to Target	$\{n\} \times \mathscr{D}^n_{x/\delta}$	$\mathscr{M}^{n-1}$	M		Landmarks	{Landmark MDP}$\cup$ {E-LSP, HIPP}
	Multi-Level Mapping **(.3.)**	$\{n\}^{\lvert\mathscr{C}_k\rvert} \times (\mathscr{D}_{x/\delta} \times \mathscr{M})^n$	$\mathscr{M}^{\lvert\mathscr{C}_k\rvert}$	M		Multi-level	HIPP,(CHLM,FDP,LSP,SNE,PLMP,LAMP)
Axes	Name Synthesized Dimension	$(\underline{\mathscr{Z}} \times \mathscr{M})^n$	$\{\lvert\underline{\mathscr{Z}}\rvert\} \times \underline{\mathscr{M}}$	U	$\mathscr{M}$	$\star$(~~MN~~)	PCA, Classical MDS, NLM, NeRV
	Map Synthesized Dim. to Original Dim.	$(\mathscr{D}_x \times \mathscr{M})^n$	$\{p\} \times \underline{\mathscr{M}}$	U	$\mathscr{D}$	Cartesian	PCA, iPCA, PCP, LAMP, PLMP, Kelp
	Discover Relation Between Original Dim.	$\{p\} \times \underline{\mathscr{M}} \times (\mathscr{D}_x \times \mathscr{M})^n$	$\subseteq \{p\}$	U	$\mathscr{D}$	Cartesian	
	Dis. Relation Btw. Visual Pattern & Orig. Dim.	$\{p\} \times \mathscr{D}_x \times (\mathscr{D}_x \times \mathscr{M})^n$	$\subseteq \{p\}$	U	$\mathscr{D}$	Cartesian	Any but (MCA)
Items Base Layout	Discover a Seed Point	$\mathscr{M}^{\lvert\underline{\mathscr{C}}_k\rvert}$	$\underline{\mathscr{C}}_k$	U	$\mathscr{M}$	$\star$(~~FN~~)	CCA, t-SNE, NeRV
	.2. Discover Neighbors in Map	$\{n\} \times \mathscr{M}^n$	$\subseteq \{\underline{n}\} \setminus f$	U	$\mathscr{M}!$	$\star$(~~FN~~)	
	.3. Brush in Map	$\mathscr{M}^g \times \mathscr{M}^n$	$\subseteq \{n\}$	U&M	$\mathscr{M}!$	$\star$(~~FN~~)	
	.4. Navigate in Map **(.3.)**	$\{\underline{n}\} \times \mathscr{M}^n$	$\underline{\mathscr{N}_{\mathscr{M}_{f_t}}}$	U	$\mathscr{M}!$	$\star$(~~FN~~)	
	Discover a Path in Map **.4.**	$\{\underline{n}\}^2 \times \mathscr{M}^n$	$\subseteq (\underline{n})$	U	$\mathscr{M}!$	$\star$(~~FN~~)	
	.5. Discover Clusters in Map **(.3.)**	$\mathscr{M}^n$	$\underline{\mathscr{C}}^n$	U	$\mathscr{M}!$	$\star$(~~FN~~)	
	Discover Outlier in Map **.2.**	$\mathscr{M}^n$	$\{\underline{n}\}$	U	$\mathscr{M}!$	$\star$(~~FN~~)	
	Dis. Sensitivity-based Outlier in Data Space	$\mathscr{M}^n \times (\mathscr{M}^{n-1})^n$	$\{\underline{n}\}$	U&M	$\mathscr{M}!$	Glob. & Lin.	PCA, Classical MDS
Items in Enriched Layout	Name Cluster 1	$(\mathscr{D}_x \times \mathscr{M})^{\lvert\underline{\mathscr{C}}_k\rvert}$	$\underline{\mathscr{L}}$	U	$\mathscr{D}$	Cart. (~~FN~~)	CCA, t-SNE, NeRV
	Name Cluster 2	$(\underline{\mathscr{Z}} \times \mathscr{M})^{\lvert\underline{\mathscr{C}}_k\rvert}$	$\underline{\mathscr{L}}$	U	$\mathscr{D}$	$\star$(~~FN~~)	
	.6. Discover Neighbors in Data Space	$\{n\} \times \mathscr{D}_\delta \times \mathscr{M}^n$	$\subseteq \{\underline{n}\} \setminus f$	U	$\mathscr{D}$	$\star$(~~MN~~)	PCA, Classical MDS, NLM, NeRV
	.7. Brush in Data Space	$\mathscr{M} \times \mathbb{R}^+ \times \mathscr{D}^n_\delta$	$\subseteq \{n\}$	U&M	$\mathscr{D}$	$\star$(~~MN~~)	
	.8. Navigate in Data Space **(.7.)**	$\{\underline{n}\} \times \mathbb{R}^+ \times (\mathscr{M} \times \mathscr{D}_\delta)^n$	$\underline{\mathscr{N}_{\mathscr{D}_{f_t}}}$	U&M	$\mathscr{D}$	$\star$(~~MN~~)	
	Discover a Path in Data Space **.8.**	$\{\underline{n}\}^2 \times \mathbb{R}^+ \times (\mathscr{M} \times \mathscr{D}_\delta)^n$	$\subseteq (\underline{n})$	U&M	$\mathscr{D}$	$\star$(~~MN~~)	
	Dis. Density-based Clusters in Data Space **.7. .8.**	$\mathbb{R}^+ \times (\mathscr{M} \times \mathscr{D}_\delta)^n$	$\underline{\mathscr{C}}^n$	U&M	$\mathscr{D}$	$\star$(~~MN~~)	
	Dis. Density-based Outlier in Data Space **.7.**	$\mathbb{R}^+ \times (\mathscr{M} \times \mathscr{D}_\delta)^n$	$\{\underline{n}\}$	U&M	$\mathscr{D}$	$\star$(~~MN~~)	
	Dis. Neighbors of OOC Item in Data Space	$\mathscr{D}_\delta \times \mathscr{M}^n$	$\subseteq \{\underline{n}\}$	U	$\mathscr{D}$	$\star$(~~MN~~)	
	Discover Class-Outlier in Data Space **(.6.)(.7.)**	$\mathscr{L} \times \mathbb{R}^+ \times (\mathscr{M} \times \mathscr{D}_\delta \times \mathscr{L})^n$	$\{\underline{n}\}$	U&M	$\mathscr{D}$	Supervised	{Supervised MDP}=
	Discover Class-Outlier in Map	$\mathscr{L} \times (\mathscr{M} \times \mathscr{L})^n$	$\{\underline{n}\}$	U	$\mathscr{M}!$	Supervised	{LDA, ClassiMap, SLLE, S-Isomap,
	.9. Match Clusters and Classes in Map **.5.**	$(\mathscr{M} \times \mathscr{L} \times \underline{\mathscr{C}})^n$	$\mathbb{R}^+$	U	$\mathscr{M}!$	Supervised	MUHSIC, S-NeRV,DS-tSNE}
	- Evaluate Cluster Purity in Map **.5.**	$\underline{\mathscr{C}} \times (\mathscr{M} \times \mathscr{L} \times \underline{\mathscr{C}})^n$	$\mathbb{R}^+$	U	$\mathscr{M}!$	Supervised	
	- Evaluate Class Compactness in Map **.5.**	$\mathscr{L} \times (\mathscr{M} \times \mathscr{L} \times \underline{\mathscr{C}})^n$	$\mathbb{R}^+$	U	$\mathscr{M}!$	Supervised	
	Classify Out-of-Core Item in Map **.2.** **(.1.)**	$\mathscr{M} \times (\mathscr{M} \times \mathscr{L})^n$	$\underline{\mathscr{L}}$	U	$\mathscr{M}!$	Supervised	
	Steer Projection ($\mathscr{Z} = \mathscr{L}$ or $\mathscr{D}_{x/\delta}$) **(.9.)**	$(\mathscr{M} \times \mathscr{Z})^n$	$\mathscr{M}^n$	U&M	$\mathscr{M}$	Steer. (~~FN~~)	LSP, PLMP, LAMP, LoCH,
	- Define Map-Landmarks Given Map **(.9.)**	$(\mathscr{M} \times \mathscr{Z})^n$	$\mathscr{M}^r$	U	$\mathscr{M}$	(~~FN~~)	Kelp, PLP, E-LSP, HIPP
	- Change Map-Landmarks' Positions	$(\mathscr{M} \times \mathscr{Z})^n \times \mathscr{M}^r$	$\mathscr{M}^r$	U	$\mathscr{M}$	(~~FN~~)	
	- Map Data Given Map-Landmarks	$(\mathscr{D}_{x/\delta} \times \mathscr{M})^n \times \mathscr{M}^r \times \mathscr{M}^r$	$\mathscr{M}^n$	M	–	-	
	Sample Data Space from Map ($\mathscr{Z} = \mathbb{R}$) **(.9.)**	$(\mathscr{M} \times \mathscr{Z})^n$	$(\mathscr{D}_x \times \mathscr{M} \times \mathscr{Z})^r$	U&M	$\mathscr{M}$	Cart. (~~FN~~)	CCA, t-SNE, NeRV
	- Define Map-Landmarks Given Map **(.9.)**	$(\mathscr{M} \times \mathscr{Z})^n$	$\mathscr{M}^r$	U	$\mathscr{M}$	(~~FN~~)	
	- Position Map-Landmarks in Data Space	$\mathscr{M}^r \times (\mathscr{D}_x \times \mathscr{M})^n$	$\mathscr{D}^r_x$	M	–	Cartesian	
	- Comp./Disp. Map-Landmarks' Values	$\mathscr{D}^r_x$	$(\mathscr{M} \times \mathscr{Z})^r$	M	–	Cartesian	

Table 4: Summary of **Generate** and **Explore** tasks, their formal description, their link to distortion types, to MDP techniques and taxonomies. The **Generate Layout** tasks take n data and possibly r landmarks from the `Input` space $\mathscr{D}$ to generate their projection into the `Output` space $\mathscr{M}$. The main **`Actor`** is the machine (M). The **Explore** tasks take the mapping $\mathscr{M}$, possibly the data $\mathscr{D}$ and other parameters as `Input`, and provide a modified map or sets of items as `Output`. The main **`Actor`** is the user (U). In the first column, some tasks are identified by numbers before their name, and other tasks are followed by these identifiers indicating they rely on these former tasks. When the identifier is between parentheses, this means the basic task is typically observed in the literature but is not necessary to achieve task. The full details are given in Supplementary Material. In columns **`Input`** and **`Output`**, a set is underlined like $\underline{\mathscr{M}}$ when it exists in the user's mind but neither in the machine nor in the layout. $\{n\}$ is a shorthand for $\{1,\ldots,n\}$ and (n) for the ordered set $(1,\ldots,n)$. When $\subseteq$ is not written, $\in$ has to be considered by default. We use set theory notation S^k for the k-ary Cartesian power of the set S, for instance $\{n\}^k$ is the set of k-tuples whose any element is taken in the set $\{1,\ldots,n\}$. In column **TS**, $\mathscr{D}$ denotes Data-space-targeted tasks; $\mathscr{M}$ denotes Map-targeted tasks; and $\mathscr{M}!$ denotes ambiguous tasks with strong risk of MDP misuse for users unaware of MDP principles and distortions. The full notation is given in the formal description of each task in the supplementary material. The **MDP property** column refers to the taxons used in Table 1. An MDP must has the corresponding property to support the task. In this column, $\star$ refers to all MDP techniques, and (~~MN~~)/(~~FN~~) means it is advisable to have few or no Missing Neighbors/False Neighbors respectively. The **Best matching MDP** column refers to the list of main MDP techniques described with references in Tables 1 and 2 best matching the tasks as discussed in Sections 2.4 and 4.3. MDP that need to be adapted are between parentheses.

shows that only the $\mathscr{D}$-tasks and the $\mathscr{M}$-tasks are *well defined* according to Munzner's typology as they provide the necessary information to be possibly achieved. Still it cannot be guaranteed the task will eventually be achieved as it also depends on all the remaining uncertainties at the visualization stage ($s7$) and on the human side stages ($h1-h10$) referring to stages in the knowledge generation model for visual analytics described in Sacha et al. [131].

The above analysis also makes clear that for the $\mathscr{M}!$-tasks, the *expressiveness principle* of visualization [103] is violated. This principle claims that the visual encoding should express all of, and only, the information necessary and sufficient to achieve the tasks. As $\mathbf{X}|\Delta$ is missing, the $\mathscr{M}!$-tasks can not be achieved, *e.g.* if a cluster or an outlier is discovered in the map, nothing can ensure it is also a cluster or an outlier in the data space.

This qualitative result has been largely overlooked so far. Fortunately, all of the $\mathscr{M}!$-tasks have a $\mathscr{D}$-task counterpart which is well defined and can be achieved based on an enriched layout. The detailed description of the tasks in the supplementary material shows that the composite tasks such as *Match Cluster and Classes*, rely on *Brush* or *Discover Neighbors* component tasks, for which exist well defined versions *Brush in Data Space* and *Discover Neighbors in Data Space* to replace their ambiguous counterparts.

How much the achievement of the $\mathscr{D}$-tasks benefits from their "data awareness", especially compared to their $\mathscr{M}!$-task counterparts, is still an open question which would require task-specific quantitative analysis with uncertainty aware and unaware users. A preliminary study conducted for ProxiViz [63] goes into that direction showing that FN and MN distortions have significantly different impacts on certain analytic tasks.

$\mathscr{D}$-tasks are based on enriched MDP layouts. We propose a survey of these enrichment techniques in Section 5. As mentioned earlier, the formal description of our task taxonomy also eases the connection to the MDP taxonomy. We provide guidelines to match tasks to MDP in the following section.

Table 5: Distribution of surveyed references. We consider the references in Section 2.3 for the **Generate Layout** tasks, and the references in Section 4, and Section 2 and Table 2 of the Supplementary Material for the **Explore** tasks. Tasks are listed in same order as in Table 4.

Tasks	#Ref
Generate Layout	
Dimension Synthesis	48
Out-of-Core Extension	8
Map Data with Intermediary Landmarks	9
Map Labeled Data	7
Map Items Relative to Target	9
Multi-Level Mapping	7
Explore Dimensions (Axes)	
Name Synthesized Dimension	5
Map Synthesized Dim. to Original Dim.	9
Discover Relation Btw. Original Dim.	4
Discover Relation Btw. Visual Pattern & Original Dim.	4
Explore Items in Base Layout	
Discover a Seed Point	1
Discover Neighbors in Map	5
Brush in Map	4
Navigate in Map	2
Discover a Path in Map	4
Discover Clusters in Map	7
Discover an Outlier in Map	3
Discover Sensitivity-based Outlier in Data Space	2
Explore Items in Enriched Layout	
Name Cluster	4
Discover Neighbors in Data Space	5
Brush in Data Space	2
Navigate in Data Space	2
Discover a Path in Data Space	2
Discover Density-based Cluster in Data Space	3
Discover Density-based Outlier in Data Space	1
Discover Neighbor of Out-of-Core Item in Data Space	1
Discover Class-Outlier in Data Space	2
Discover Class-Outlier in Map	1
Match Clusters and Classes in Map	14
Classify Out-of-Sample Item in Map	2
Steer Projection by Moving Landmarks in Map	11
Sample Data Space from Map	2

4.3 Which is the Best MDP for an Intended Task?

The formal description of the tasks in our taxonomy and the taxons of the MDP taxonomy from Section 2.3 enable us to identify which MDP techniques are best suited for an intended task.

As mentioned in the MDP taxonomy, the taxons are: data types; linearity; supervision; multi-level; locality; steerability; stability; out of core extension (OOC); and computation burden (Comp). We report this taxon in the *MDP property* column of Table 4.

All MDP techniques achieve the **Generate Layout** tasks, synthesizing original dimensions. Still only some of them can handle *Out-of-Core Extension* which are directly reported in Table 1 column OOC. Landmark and control point-based techniques such as L-MDS, Pivot-MDS, PLP, LAMP implement *Map Data with Landmarks*. The supervised techniques reported in Table 1 column Supervision are designed to *Map Labeled Data*. Beyond specific techniques designed to *Map Items Relative to Target*, this task can be implemented by any landmark/control point-based technique, using the target item as one of the landmarks kept fixed in the map, only updating the position of the other items. *Local Mapping* can also be achieved by hierarchical MDP techniques (column Multi-level), besides obviously the local techniques with the local mapping property (see Section 2.3.5). The full list of MDP related to each *Generate Layout* tasks is reported in the Table 4.

Regarding **Explore Dimensions (Axes)** tasks mentioning *Original Dimension*, and all tasks indicating $\mathscr{D}_x$ as `input`, as *Name Cluster* and

Sample Data Space from Map, require MDP able to handle *Cartesian* coordinates.

Tasks indicating $\mathscr{L}$ as `Input` like the ones mentioning *Class-Outlier*, *Class Compactness* or *Cluster Purity*, and the *Match Clusters and Classes in Map* and *Classify Out-of-Core Item in Map* tasks, are best handled by *Supervised* MDP. Notice that they are part of the *Items in Enriched Layout* so the class is actually overlaid (typically as a color or text), hence visible in the MDP layout.

Explore Items in Base Layout tasks, which are only focused on the map itself (indicated by $\mathscr{M}$ or $\mathscr{M}!$ as target space (TS)), can be theoretically supported by any MDP technique because they do not depend on the mapping quality. Still, for improved effectiveness, if no other intended task imposes a more specific MDP, we can advise to use an MDP which preserves trustworthiness rather than continuity, *i.e.* which generates few or no False Neighbors (FN) distortions, even if it can generate Missed Neighbors (MN). The rationale behind this rule is that the *gestalt law of proximity* makes point patterns like clusters in the map pop up pre-attentively. Therefore it is better that the items seemingly part of the same pattern because they are neighbors in the layout, are actually neighbors in the data space too, which is possible with MN, but not possible with FN. In short, we need to limit FN to ensure the proximity-pattern we see in the MDP layout with no effort, actually exist in the data space. This means we shall prefer MDP techniques whose energy function has the *plasticity* property as defined by Lee and Verleysen [85] such as CCA, t-SNE and NeRV with the CCA-like parameter setting, or by lack of theoretical hints, MDP layouts that have empirically better *trustworthiness* scores. The task *Name Cluster* would also benefit from such MDP

Explore Items in Enriched Layout tasks, which are focused on the data analysis through their map representation (indicated by $\mathscr{D}$ as target space (TS)), can be supported by any MDP technique which preserves continuity rather than trustworthiness, *i.e.* which generates few or no Missed Neighbors (MN) distortions, even if it can generate False Neighbors (FN). Assuming the color channel is used for the layout enrichment, the rationale behind this rule is that the *gestalt law of similarity* makes color patterns pop up pre-attentively. Therefore, it is better that the items seemingly part of the same data-pattern because they have the same color in the layout, are actually neighbors in the map too, which is possible with FN, but not possible with MN. In short, we need to limit MN to ensure the color-pattern we can see with no effort in the MDP layout corresponds to a single continuous pattern in the data. This means we shall prefer MDP techniques whose energy function has the *elasticity* property as defined by Lee and Verleysen [85] such as PCA, NLM, Classical MDS and NeRV with the NLM-like parameter setting, or by lack of theoretical hints, MDP layouts that have empirically better *continuity* scores. The task *Name Synthesized Dimension* would also benefit from such MDP.

The task *Discover Sensitivity-based Outlier in Data Space* requires an MDP with *Global* and *Linear* mapping as an item is detected as outlier if the map appears significantly different with it from without it. The task *Steer Projection* can only be handled by MDP having the *Steerability* property. These two tasks require fast computation of the map as interactivity is key, either when moving the control points or switching a point on and off. LAMP is the best suited in that case.

High *stability* preserving context and mental map, low *computational cost* enabling interactivity, and the *multi-level* property implementing the Schneiderman mantra *overview first, zoom and filter, then details-on-demand*, are generic desirable properties for any intended task. By default, everything else being equal, MDP having these properties have to be considered first.

The list of MDP recommended for each intended task based on this qualitative analysis, is reported in Table 4. Still, more quantitative analyses are missing to better support data scientists using MDP.

5 Layout Enrichment

The layout resulting from a MDP mapping consists basically of a point cloud where well defined groups and neighborhoods are indicative of similarity among the involved instances. However, point proximity information is usually not enough to boost a prolific analytical process. Important aspects such as which attributes most contribute to the neighborhood formation and which is the gist information shared by a group of instances cannot be inferred from point proximity only. Moreover, as discussed previously, the awareness about errors and distortions in the projection introduces uncertainties in the analytical process and lessens the trustworthiness in the observed neighborhoods.

Layout enrichment has been an alternative to circumvent those issues, rendering projection layouts more reliable and informative. First attempts to enrich MDP layouts relied on simple textual and color information overlaid onto layouts produced by PCA and SOM [81, 172]. More recent approaches enable enrichment resources to reveal distortions [4] and rely on sophisticated data summarization schemes that facilitates the analytical process [119].

Existing layout enrichment techniques vary considerably as to the source of information used to trim the layout. Specifically, the information used to "adorn" a layout can stem from the original space, from the visual space, or from both. For instance, some methods can simply color points in the visual space based on particular attribute values of the original data, not relying in any information from the visual space (besides point location) to make the enrichment. On the other hand, there are methods that operates exclusively in the MDP layout, highlighting, for instance, clusters of projected points, thus disregarding the original data altogether. However, due to the greater flexibility, the vast majority of layout enrichment methods opt to play with information from both spaces, typically mapping content from the original data to points and/or neighborhoods in the visual space.

Enrichment methods can also be characterized based on where additional visual resources are placed, i.e., overlaid on the point cloud itself or externally in a panel separated from the projection layout. Techniques that use overlaid information benefit from the interplay between the gestalt laws of proximity (points position) and similarity (shape, text, color overlay) to leverage the perception of visual correlation between neighborhood structures and the extra visual content depicted within the projection layout. External enrichment methods rely on visualization panels located independently from the scatter plot layout, but usually linked to it through interactive mechanisms.

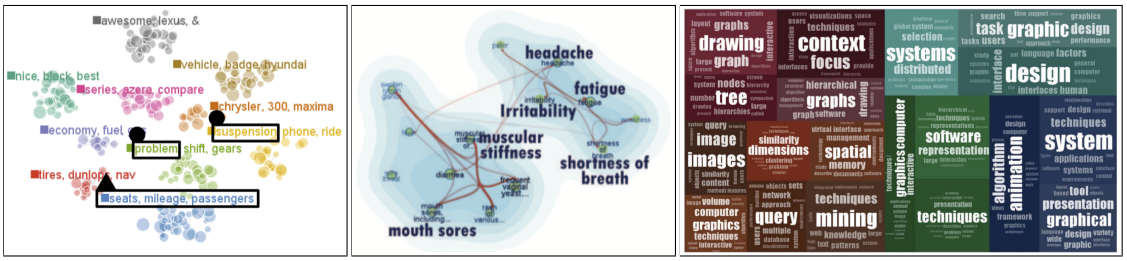
A third option is to characterize enrichment methods according to their end goal, i.e., visualize and uncover phenomena related to distortions and projection errors or enrich layouts so as to reveal gist information contained or related to the original data. In the following, we discuss enrichment approaches grouping them according to this third option, since it is simpler and allows for clearly discriminating methodologies. It is important to emphasize that our intention here is not to list all visualization methods that somehow rely on enrichment schemes to improve MDP layouts, but rather propose a taxonomy for the enrichment strategies, providing examples of where such taxonomy have been employed.

5.1 Enrichment for Content Analysis

Content-based enrichment techniques build upon the proximity of similar instances in the visual space to depict additional information associated to particular instances or groups of instances. The manner data content is depicted varies among techniques and target application, but, in a broad sense, the content is always a summary of gist information extracted from the original data. Further, content-based enrichment methods can be grouped according to the pipeline they rely on, which are basically of three types: Direct Enrichment, Cluster-based Enrichment, and Spatially Structured Enrichment, as illustrated in Figure 2.

5.1.1 Direct Enrichment

Direct Enrichment methods enhance the layout directly, without segmenting or partitioning the visual space. A common approach is to enrich the MDP layout by combining overlaid tags extracted from attributes [60] with external linked views where additional information is presented [37, 38, 82] (see Figure 2(a)). More recent approaches are able to map instances and attributes in a single visual space while preserving their semantic relation, resulting in a layout composed by dots and tags that represent instances and attributes respectively [18, 36].



(a) Direct Enrichment [37] (b) Cluster-based Enrichment [31] (c) Spatially Structured Enrichment [46]

Fig. 2: Layout enrichment for content analysis (permission to use the images will be requested to IEEE).

Representing projected instances by icons [30] or geometric objects [58] rather than points is also a recurrent alternative commonly combined with overlap removal and object alignment to improve layout readability [57]. Some direct methods seek to reduce cognitive effort by rendering MDP layouts as physically plausible environments, aiming to mimic an explorable scenario such as galaxies [3] and cartographic maps [146].

Color coding points according to particular attributes is a basic enrichment scheme [163] called *Component Plane*, which is used to support more sophisticated alternatives. For instance, generalized biplots [39] extend and enhance biplots to show data attributes in a three-dimensional visual space, using interactive bar charts and color legends to highlight variables better visible from each viewpoint. Still envisioning to understanding attributes and their impact in a scatter plot formation, some approaches propose enrichment mechanisms to color the Delaunay triangulation of mapped points according to information extracted from their original attributes [142]. Node-link and edge bundling can also be incorporated to allow the visualization of temporal information associated to the data [143]. Color codes have also been used to depict the distance between any data item to a new item not mapped to the visual space [6], allowing the visual classification/inspection of new data without the need of projecting it.

5.1.2 Cluster-based Enrichment

Methods that rely on Cluster-based Enrichment typically cluster points according to their proximity in the visual space, generating well defined and non-overlapping regions in the layout. The enrichment is then accomplished by depicting content computed from instances in each clusters. For instance, in the context of textual data visualization, a typical approach is to summarize the content of each cluster as a word cloud rendered inside the cluster [119, 174]. Nevertheless, the enrichment process can reach a high degree of sophistication, adorning clusters with content aware objects such as time series plots [149], images [52], and tailored icons [30]. Some system can even resort to bundled lines to highlight relations between instances in different clusters, linking the scatter plot layout with external panels containing meaningful information about the content of each cluster [31] (Figure 2(b)).

Most cluster-based enrichment methods perform the clustering step in the visual space without considering the distortion introduced during the MDP process. Although the visual space-based clustering guarantees non-fragmented groups in the layout, it increases uncertainty in what is being analyzed, since distortions, in particular False Neighbors, are not typically taken into consideration when extracting information to be depicted. This problem has been attenuated by first selecting representative points in the original space and then clustering the remaining instances around those points in the visual space [74]. The U-matrix [94] is another alternative that uses a Self-Organizing Map with a large number of prototypes (possibly more than the data items). The many prototypes which end up encoding no data, lying in the empty space between data clusters, are used to show the boundary of the clusters on the map.

There are also techniques that perform the clustering in the data space, highlighting their content in the visual space [45, 82, 104]. The problem with those methods (as for the U-matrix), is that clusters can appear fragmented in the visual space due to MN distortions, increasing the cognitive effort to interpret the layout.

5.1.3 Spatially Structured Enrichment

Similarly to Cluster-based Enrichment methods, Spatially Structured Enrichment techniques operate in two stages, first organizing projected data in a geometric structure and then performing the enrichment. These techniques count on some density estimation in the visual space to guide a space partition scheme such as Voronoi tesselations [27, 105, 149], Treemaps [46] (Figure 2(c)), Quadtrees [56] or simply a regular grid [107]. Enrichment is then carried out on the regions defined by the partition, ranging from word clouds [46] to visual snippets [56]. Although these methods also bear the issue of computing the partition in the visual space without considering information from the original space, thus being sensitive to MN and FN distortions, most of them naturally provide a hierarchical navigation mechanisms that favor exploratory tasks.

By properly combining attributes and their correlation in a single data matrix, the approach implemented in Data Context Map [36] takes into account information in the visual space as well as from the data space to build a density function whose level sets segment the layout according to the original attributes and their range of values.

Although not so abundant, there are techniques that comprise more than one of the approaches above, as is the case of Probing Projections [148], which enable resources for direct and cluster-based enrichment.

5.2 Enrichment for Distortion Analysis

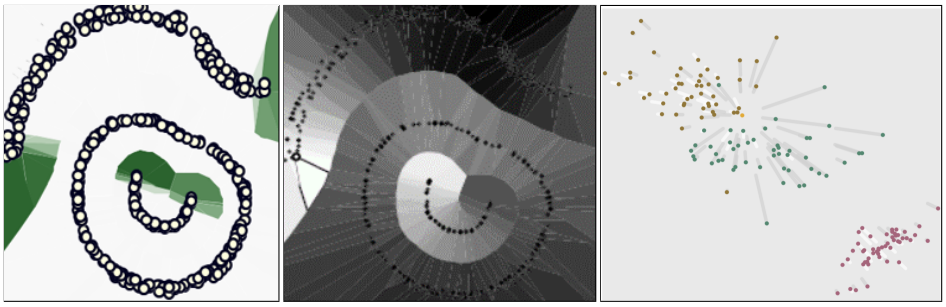
Regarding the uncertainty typology introduced by Sacha et al. [130], if we assume a high effectiveness of the scatter plot idiom used to visualize the data, and a high ability of users to read the map, then the main source of uncertainties comes from the MDP distortions. Several mechanisms have been proposed to visualize distortions introduced during the MDP process so as to make users aware of the interpretation pitfalls. They can at best be used to identify trustworthy areas of the map, and even effectively detect patterns in the data thanks to displayed distortions.

Enrichment methods for distortion analysis also vary considerably as to graphical and interactive infrastructure employed in the enrichment process, ranging from simple mechanisms as Shepard diagrams [73] to sophisticated interactive system able to reveal multiple facets of the MDP distortions [97].

In the following, we organize enrichment mechanisms for distortion analysis into two categories, visualizing trustworthy and unreliable regions in the visual space and probing mechanisms to further understanding distortions. Figure 3 depicts examples of enriched layouts in each category.

5.2.1 Visualizing Trustworthy and Unreliable Regions

The simplest enrichment mechanism for distortion analysis is to color-code projected points according to some distortion measure [100], enabling additional exploratory resources via linked external panels [93]. However, many techniques rely on domain decomposition schemes



(a) CheckViz Enrichment [89] (b) ProxiViz Enrichment [4] (c) Probing Projections [148]

Fig. 3: Layout enrichment for distortion analysis (Permission to use image (c) will be requested to IEEE).

to partition the visual space in order to highlight regions with high and low distortions. A natural scenario for this kind of analysis is SOM-based MDP, as distortion can be measured in each SOM cell and color coded in the visual space [75]. Landscape metaphor depicting distortions as a height field is a variant that generalizes to any MDP method [137] [166]. Color-coding Voronoi cells or Delaunay edges of projected points to highlight regions of high and low distortions is another space partitioning alternative [4]. All these techniques tell which parts of the map are more trustworthy, so where patterns (outliers, lines, clusters) in the map can be inferred to exist in the original data space too. However, parts of the map considered as not trustworthy must be used with caution to avoid erroneous inference, or shall be used only as a proxy of original data patterns to generate hypotheses to be verified using complementary trustworthy views.

More sophisticated color-code schemes are able to simultaneously highlight more than one distortion phenomenon simultaneously. This is the case of CheckViz [89], where a bicolor colormap is used to code the missing and false neighbor information around projected points (see Figure 3(a)). CheckViz enables the inference of patterns beyond the simple analysis of distortions and detection of trustworthy areas, leveraging the identification of data patterns like *true class-overlap* and *true cluster-separation* directly in the mapping.

Beyond these color schemes, small-multiples has also been used to show at each point in the layout a small clone of the full map as a snippet image [128]. The snippet images are color-coded so as to reveal both False and Missing Neighbors.

A standard scheme to analyze distortions in labeled data is to color-code the data with their class label. The aim is to use the *Match Clusters and Classes* task assuming that classes form pure clusters in the data space. When they do not match, *i.e.* clusters appear not pure or classes are split in several components, the map is assumed erroneous [5].

Alternative ways to this evaluation approach have been proposed [5], where classes detected as overlapping in the data space are first merged, or a single one of them is used and the other removed. Graph-based schemes [7, 51, 90] have also been proposed to evaluate class overlap in the data space first and then the result used as a baseline for the color-coded map evaluation.

Color code and domain decomposition has been employed by Aupetit [4, 6] as a way to check the quality of the projected out-of-core data, enriching the layout according to the similarity between OOC and core data.

5.2.2 Distortion Analysis via Probing

Techniques that rely on probing mechanism enable interactive resources for users to inspect particular regions of the visual space in order to further understand how distortions are taking place in those regions. ProxiViz [4] is based on selecting a point in the map, and color-coding the Voronoi cells of the points according to their dissimilarity in the data space with respect to the selected point (see Figure 3(b)). It is implemented in the VisCoDeR application [40] to support on-line self-teaching of MDP techniques. Its variant Proxilens [62] enables an interactive lens-like resource from which users can filter out false neighbors around inspected points while visualizing missing neighbors through a background color scheme. Probing Projections [148] displays a halo around each point to depict error (see Figure 3(c)). A probing mechanism allows users to further understand distortions by selecting a particular point around which the system remove distance errors by rescaling the vector between the selected and neighbor points according to distance error ratio. A more elaborated mechanism is proposed by Martins et al. [97], which color code Delaunay triangles built from the projected point cloud according to False Neighbor information, allowing users to select particular regions in order to reveal Missing Neighbors using bundled linked edges.

A different enrichment methodology for distortion analysis has been proposed in ProjInspector [108]. In contrast to the methods described above, ProjInspector does not enrich a layouts to evaluate a single projection but rather to compare multiple projections in order to identify the most accurate ones, enabling an external panel where the error from different projections are simultaneously analyzed. ProjInspector can also interpolate layouts in order to generate new projections.

5.3 Which Layout Enrichment to Use?

In this section we discuss some characteristics of enhancement mechanisms, limitations, and scenarios of usability.

Direct enrichment for content analysis techniques are straightforward in the sense that the layout produced by them are simple and easy to interpret, not demanding a high cognitive effort from users. However, this simplicity brings limitations. For instance, when tags and colors are used to convey the content of the data, it might become difficult to figure out which instances are contributing to each color or tag. Moreover, the optimal position of tags, glyph shape, and color map is an issue to be tackled. Therefore, in many applications, direct enrichment is mainly used as a mechanism to inspect particular instances or group of instances, thus being combined with interactive resources.

Similarly to direct enrichment, **cluster-based enrichment** schemes are easy to interpret, have the advantage that one can clearly figure out which data items are related to the highlighted information, as those information are typically depicted within the clusters. Moreover, this type of enrichment can effectively be employed in applications involving large data sets, since what really matters in the enrichment process is the cluster configuration. However, depicting information within small or geometrically intricate clusters is an issue, rendering cluster-based methods not appropriate for applications where the granularity of the clusters is high. In other words, cluster-based techniques are useful to convey a summary of the data content, but not so effective to depict local information associated to a large number of small sets of items.

In contrast, **spatially structured enrichment** can handle large amount of data while being able to depict information associated to a reasonably large number of groups. The reason is that the visual space is partitioned in regularly shaped objects such as spheres and rectangles, making easier to place the information with respect to objects. However, the resulting visualization tends to be dense, demanding a greater cognitive effort to interpret it. Another issue is that the projected point cloud can become difficult to visualize and interact with, making

spatially structured enrichment not so adequate to applications where particular neighborhoods have to be explored.

Regarding **enrichment for distortion analysis**, techniques based on color-coded error measurements figure among the simplest and easier to interpret. However, special caution should be taken when color-coding dissimilarity error types, as distances in the original and visual space do not match due to the shift phenomena [85], demanding some preprocessing such as scaling and normalization of values before employing them as enrichment resource. When dealing with label data, the widely used scheme of color-coding the projected points according to class labels to visually validate the mapping might lead to wrong conclusions. The class-as-pure-cluster assumption is practically never validated in the data space before use. Not checking the validity of this assumption may lead to the erroneous inference that classes overlap in the data space when they overlap in the map. One way to come over this is to check for actual data neighborhood in the data space using a secondary layout enrichment, bearing this information like the proximity view [4]. Regarding MDP evaluation, assuming erroneously that classes are well separated in the data space and seeing them overlapping in the map leads to an overly pessimistic evaluation of the map quality: the classes are said to overlap because of a bad mapping, while in fact they visually overlap because they actually overlap in the data space. First checking with data analysis techniques the actual class separation in the data space can be a way to solve this issue [5].

Probing-based enrichment schemes are also easy to interpret and typically provides a clear picture of the distortion in the analyzed neighborhood, still not immune to the shift-phenomena as for color-coded enrichment for distortions analysis. However, probe scheme would benefit from being combined with color-coded global distortions in order to indicate to users which are the "interesting" regions to probe. In other words, probing mechanism should be implemented on the top of trustworthy and reliable enrichment schemes, without which the use of probing mechanism can become an overwhelming task.

6 Future and Open Questions

Improving MDP techniques From Table 1 one can identify several scenarios and directions where MDP techniques can evolve. For instance, it is clear that there is no MDP method able to handle data from all types. In particular, more work is needed towards developing MDP methods able to handle categorical data while bearing properties interesting to visualization applications such as *steerability* and *stability*.

Another interesting question is which other properties beyond the ones listed in Table 1 are fundamental for visualization applications that have not yet been implemented in the MDP methods. Which are the ideal properties for visualization applications?

Some recent work [164, 176] propose to map attributes as points instead of the data items. There are even techniques that modify the data so as to consider simultaneously data items and attribute correlation as input data [36]. How to properly exploit the interplay between attribute maps and data maps is an open issue with a large range of opportunities.

Improving MDP layout enrichment Enriching the map with probability information could benefit probabilistic MDP methods such as NeRV and t-SNE. Very few has been done to enable enriched layout to assist the interpretation of those important MDP methods.

The curse of dimensionality has mainly been handled on the theoretical side by studying the shift-invariant property [85] of NeRV and t-SNE, in order to design new MDP techniques more robust to this phenomenon. Another more empirical approach used in a multidimensional brushing technique based on a linear MDP [8], consists in setting interactively the size of the spherical brush by comparing the empirical distribution of the distances to its center, to the theoretical distribution of the same if the points were normally distributed. The remnant color let by the brush, acts as a progressive probing-based layout enrichment which takes into account the shift-invariant property. Despite these two approaches, this key property is far from having been fully exploited to enrich layouts towards better assisting in map evaluation, and better guiding in MDP-based multidimensional data exploration.

Maximizing effectiveness of MDP layouts Concerning uncertainties at the Visualization stage described in Section 3.1.1, there is a lack of empirical evidence regarding how scatter plots specifically produced by MDP, as they bear no meaningful direction, should be displayed within a circular or a square frame of reference [109] to maximize its effectiveness.

Moreover, although studies exist regarding the perception of correlation [124] or of aggregated measures [54] in scatter plots, and taxonomies of color-coded point patterns have been proposed [136], we are not aware of visual perception studies regarding how the distance between two points in the MDP layout is perceived especially when embedded in cloud of points forming some shape. Beyond assuming that the Euclidean distance mimics at best the proximity perception in the layout, such studies could give insight about how to measure the distances in the layout from a perceptual point of view to best encode original similarities therein, like it has been achieved with the study of perceptually uniform color spaces. Such studies could give stronger support to heuristic guidelines for rendering MDP layouts and could possibly lead to the design of more perceptually effective MDP techniques.

Leveraging on the taxonomies of MDP analytic tasks and distortions In this work, we formalize many MDP-based analytic tasks. Our taxonomy and seminal formalization of MDP-based analytic tasks also call for a series of quantitative user-studies to evaluate how mapping distortions effectively impact user's ability to achieve a specific task. This would allow validating or invalidating the qualitative results presented in Section 4.2, providing more specific quantitative information, leading to user guidelines better grounded than empirical [168] or qualitative ones as provided in Table 4. The case of the ambiguous MDP-based analytic tasks ($\mathscr{M}$!) discovered thanks to their formal description in the task taxonomy, advocates for generalizing such formalization to other tasks and idioms by their designers so to debunk possibly inexpressive visualizations beforehand.

In line with theoretical work regarding the visualization pipeline [35, 130, 131], we also encourage to explore this direction further to put these types of formal descriptions into a larger framework that would make clear the information at stake in the visual analytic pipeline. It would also allow better supporting the design of complex human-in-the-loop visual analytic systems.

We also show that there are mainly two kinds of distortions: Missing Neighbors and False Neighbors. Preliminary studies [63] uncovered that these distortions do not impact the same way data analysis tasks. Other recent study [90] exploits different types of distortions to generate supervised mappings that better preserve the class structure of the original labeled data. We could generalize this type of study to determine for each MDP technique the type of distortions it mainly generates. Knowing the relation MDP technique versus distortions and for each MDP-based analytic task the distortions it is better to have (if they cannot be avoided), one could refine our guidelines to recommend which is the optimal MDP technique for a given analytic task.

This approach comes with a recent trend in visualization to develop *visual quality metrics* [20] specifically for scatterplots [9, 112, 135], which can be used to automatize the search for good quality visualizations to recommend to the user [41, 78]. We could use such techniques to recommend an MDP layout optimal for a certain task among a huge set of parametric visualizations, finding the best parameters to encode the data so that the user rapidly spot the pattern of interest to him/her among a huge set of possibilities.

MDP as a Support for Visualization assisted Machine Learning We are witnessing an astonishing growth of visualization techniques tailored to support machine learning tasks and MDP has been playing a fundamental role in this context. In fact, MDP has been the basis for visualization-assisted systems devoted to metric learning [28, 95], active learning annotation [68], and deep neural network model understanding [123]. We are convinced that MDP will become one of the main visualization tools in this scenario, and a multitude of possibilities are still open to be explored.

ACKNOWLEDGMENT

We are grateful to the reviewers for their very thoughtful comments. M.A thanks John A. Lee for co-leading the working group "Embedding techniques at the crossing of Machine Learning and Information Visualization" at the 2012 Dagstuhl seminar 12081 [76], and Laurens van der Maaten for co-organizing the EuroVis 2013 Workshop on Visual Analytics using Multidimensional Projections [11] and its special issue [12], which all contributed to fruitful discussions on the use of MDP in Visual Analytics. M.A. also would like to thank Nicolas Heulot, Sylvain Lespinats and Jean-Daniel Fekete for many years of collaborative work on designing new supervised MDP and interactive enrichment techniques which contributed to a deeper understanding of the distortions in MDP. We also thank Dominik Sacha and Michael Sedlmair who accepted to share with us the papers they used in their survey [131] saving us a huge time in gathering the relevant material.

Grants 302643/2013-3 CNPq-Brazil and 2016/04391-2 São Paulo Research Foundation (FAPESP) - Brazil. The views expressed are those of the authors and do not reflect the official policy or position of the São Paulo Research Foundation.

REFERENCES

[1] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Symp. Info. Vis.*, pp. 111–117, 2005.

[2] E. Amorim, E. Brazil, L. G. Nonato, F. Samavati, and M. Sousa. Multidimensional projection with radial basis function and control points selection. In *PacificVis*, pp. 209–216, 2014.

[3] K. Andrews, W. Kienreich, V. Sabol, J. Becker, G. Droschl, F. Kappe, M. Granitzer, P. Auer, and K. Tochtermann. The infosky visual explorer: exploiting hierarchical structure and document similarities. *Info. Vis.*, 1(3-4):166–181, 2002.

[4] M. Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7):1304–1330, 2007.

[5] M. Aupetit. Sanity check for class-coloring-based evaluation of dimension reduction techniques. In *BELIV*, pp. 134–141, 2014.

[6] M. Aupetit, L. Allano, I. Espagnon, and G. Sannie. Visual analytics to check marine containers in the eritr@c project. In *EuroVAST*, 2010.

[7] M. Aupetit and T. Catz. High-dimensional labeled data analysis with topology representing graphs. *Neurocomputing*, 63:139–169, 2005.

[8] M. Aupetit, N. Heulot, and J. Fekete. A multidimensional brush for scatterplot data analytics. In *IEEE VAST*, pp. 221–222, 2014.

[9] M. Aupetit and M. Sedlmair. Sepme: 2002 new visual separation measures. In *PacificVis*, pp. 1–8, 2016.

[10] M. Aupetit, E. Ullah, R. Rawi, and H. Bensmail. A design study to identify inconsistencies in kinship information: The case of the 1000 genomes project. In *PacificVis*, pp. 254–258, 2016.

[11] M. Aupetit and L. van der Maaten. Eurovis workshop on visual analytics using multidimensional projections. *https://diglib.eg.org/handle/10.2312/1001 (visited in May 2018)*, 2013.

[12] M. Aupetit and L. van der Maaten. Neurocomputing special issue on visual analytics using multidimensional projections. *https://www.sciencedirect.com/journal/neurocomputing/vol/150 (visited in May 2018)*, 2015.

[13] A. Barbosa, F. Paulovich, A. Paiva, S. Goldenstein, F. Petronetto, and L. Nonato. Visualizing and interacting with kernelized data. *IEEE Trans. Vis. Comp. Graph.*, 22(3):1314–1325, 2016.

[14] H.-U. Bauer and K. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Trans. Neural Networks*, 3(4):570–579, 1992.

[15] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[16] Y. Bengio, J.-f. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *NIPS*, pp. 177–184, 2004.

[17] J.-P. Benzécri. *L'analyse des données*, vol. 2. Dunod Paris, 1973.

[18] M. Berger, K. McDonough, and L. Seversky. cite2vec: Citation-driven document exploration via word embeddings. *IEEE Trans. Vis. Comp. Graph.*, 23(1):691–700, 2017.

[19] J. Bertin. *Semiology of Graphics*. Univ. Wisc. Press, 1983.

[20] E. Bertini, A. Tatu, and D. Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Trans. Vis. Comp. Graph.*, 17(12):2203–2212, 2011.

[21] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2016.

[22] C. Bishop, M. Svensén, and C. Williams. Gtm: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[23] I. Borg and P. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[24] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *Graph Drawing*, pp. 42–53, 2006.

[25] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Trans. Vis. Comp. Graph.*, 19(12):2376–2385, 2013.

[26] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner. Visualizing dimensionally-reduced data: interviews with analysts and a characterization of task sequences. In *BELIV*, pp. 1–8, 2014.

[27] B. Broeksema, A. Telea, and T. Baudel. Visual analysis of multi-dimensional categorical data sets. *Comp. Graph. Forum.*, 32(8):158–169, 2013.

[28] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *IEEE VAST*, pp. 83–92, 2012.

[29] A. Buja, D. Swayne, M. Littman, N. Dean, H. Hofmann, and L. Chen. Data visualization with multidimensional scaling. *J. Comp. and Graph. Stat.*, 17(2):444–472, 2008.

[30] N. Cao, D. Gotz, J. Sun, and H. Qu. DICON: Interactive visual analysis of multidimensional clusters. *IEEE Trans. Vis. Comp. Graph.*, 17(12):2581–2590, 2011.

[31] N. Cao, J. Sun, Y.-R. Lin, D. Gotz, S. Liu, and H. Qu. Facetatlas: Multifaceted visualization for rich text corpora. *IEEE Trans. Vis. Comp. Graph.*, 16(6):1172–1181, 2010.

[32] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *IEEE Visualization*, pp. 127–131, 1996.

[33] H. Chen, S. Zhang, W. Chen, H. Mei, J. Zhang, A. Mercer, R. Liang, , and H. Qu. Uncertainty-aware multidimensional ensemble data visualization and exploration. *IEEE Trans. Vis. Comp. Graph.*, 2015.

[34] L. Chen and A. Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *J. Amer. Stat. Assoc.*, 104(485):209–219, 2009.

[35] M. Chen and A. Golan. What may visualization processes optimize? *IEEE Trans. Vis. Comp. Graph.*, 22(12):2619–2632, 2016.

[36] S. Cheng and K. Mueller. The data context map: Fusing data and attributes into a unified display. *IEEE Trans. Vis. Comp. Graph.*, 22(1):121–130, 2016.

[37] J. Choo, C. Lee, C. Reddy, and H. Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Trans. Vis. Comp. Graph.*, 19(12):1992–2001, 2013.

[38] J. Choo, H. Lee, J. Kihm, and H. Park. ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *IEEE VAST*, pp. 27–34, 2010.

[39] D. Coimbra, R. Martins, T. Neves, A. Telea, and F. Paulovich. Explaining three-dimensional dimensionality reduction plots. *Info. Vis.*, 15(2):154–172, 2016.

[40] R. Cutura, S. Holzer, M. Aupetit, and M. Sedlmair. Viscoder: A tool for visually comparing dimensionality reduction algorithms. *ESANN*, 2018.

[41] T. Dang and L. Wilkinson. Scagexplorer: Exploring scatterplots by their scagnostics. In *PacificVis*, pp. 73–80, 2014.

[42] J. De Leeuw and W. Heiser. Multidimensional scaling with restrictions on the configuration. *Multivariate Analysis*, 5:501–522, 1980.

[43] P. Demartines and J. Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Trans. on Neural Networks*, 8(1):148–154, 1997.

[44] L. Di Caro, V. Frias-Martinez, and E. Frias-Martinez. Analyzing the role of dimension arrangement for data visualization in radviz. In *Advances in Knowledge Discovery and Data Mining*, pp. 125–132. Springer, 2010.

[45] C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Int. Conf. Mach. Learn.*, pp. 521–528, 2007.

[46] F. Duarte, F. Sikansi, F. Fatore, S. Fadel, and F. Paulovich. Nmap: A novel neighborhood preservation space-filling algorithm. *IEEE Trans. Vis. Comp. Graph.*, 20(12):2063–2071, 2014.

[47] S. Fadel, F. Fatore, F. Duarte, and F. Paulovich. Loch: A neighborhood-based multidimensional projection technique for high-dimensional sparse

spaces. *Neurocomputing*, 150:546–556, 2015.

[48] C. Faloutsos and K. Lin. Fastmap: A fast algorithm for indexing, datamining and visualization of traditional and multimedia databases. In *ACM SIGMOD*, pp. 163–174, 1995.

[49] F. Fernández, M. Verleysen, J. Lee, and I. Blanco. Stability comparison of dimensionality reduction techniques attending to data and parameter variations. In *EuroVis (Short Paper)*, 2013.

[50] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[51] P. Gaillard, M. Aupetit, and G. Govaert. Learning topology of a labeled data set with the supervised generative gaussian graph. *Neurocomputing*, 71(7-9):1283–1299, 2008.

[52] E. Gansner, Y. Hu, and S. North. Interactive visualization of streaming text data with dynamic maps. *J. Graph Alg. Appl.*, 17(4):515–540, 2013.

[53] X. Geng, D.-C. Zhan, and Z.-H. Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Trans. Sys., Man, and Cyber.*, 35(6):1098–1107, 2005.

[54] M. Gleicher, M. Correll, C. Nothelfer, and S. Franconeri. Perception of average value in multiclass scatterplots. *IEEE Trans. Vis. Comp. Graph.*, 19(12):2316–2325, 2013.

[55] Y. Goldberg and Y. Ritov. Local procrustes for manifold embedding: a measure of embedding quality and embedding algorithms. *Mach. Learn.*, 77(1):1–25, 2009.

[56] E. Gomez-Nieto, W. Casaca, D. Motta, I. Hartmann, G. Taubin, and L. G. Nonato. Dealing with multiple requirements in geometric arrangements. *IEEE Trans. Vis. Comp. Graph.*, 22(3):1223–1235, 2016.

[57] E. Gomez-Nieto, W. Casaca, L. G. Nonato, and G. Taubin. Mixed integer optimization for layout arrangement. In *Sibgrapi*, pp. 115–122, 2013.

[58] E. Gomez-Nieto, F. San Roman, P. Pagliosa, W. Casaca, E. Helou, M. Oliveira, and L. Nonato. Similarity preserving snippet-based visualization of web search results. *IEEE Trans. Vis. Comp. Graph.*, 20(3):457–470, 2014.

[59] A. Gracia, S. González, V. Robles, and E. Menasalvas. A methodology to compare dimensionality reduction algorithms in terms of loss of quality. *Info. Scie.*, 270:1–27, 2014.

[60] B. Gretarsson, J. O'Donovan, S. Bostandjiev, T. Höllerer, A. Asuncion, D. Newman, and P. Smyth. Topicnets: Visual analysis of large text corpora with topic modeling. *ACM Trans. Intell. Syst. Techn.*, 3(2):1–26, 2012.

[61] Y. Guo, T. Hastie, and R. Tibshirani. Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1):86–100, 2007.

[62] N. Heulot, M. Aupetit, and J. Fekete. Proxilens: Interactive exploration of high-dimensional data using projections. In *EuroVis Workshop on Vis. Anal. Using Multid. Proj.*, 2013.

[63] N. Heulot, J. Fekete, and M. Aupetit. Visualizing dimensionality reduction artifacts: An evaluation. Technical report, May 2015.

[64] G. Hinton and S. Roweis. Stochastic neighbor embedding. In *NIPS*, pp. 833–840, 2002.

[65] H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. Educat. Psych.*, 24(6):417, 1933.

[66] L. House, S. Leman, and C. Han. Bayesian visual analytics: Bava. *Stat. Anal. and Data Min.*, 8(1):1–13, 2015.

[67] X. Hu, L. Bradel, D. Maiti, L. House, C. North, and S. Leman. Semantics of directly manipulating spatializations. *IEEE Trans. Vis. Comp. Graph.*, 19(12):2052–2059, 2013.

[68] L. Huang, S. Matwin, E. Carvalho, and R. Minghim. Active learning with visualization for text data. In *ACM Work. Explor. Search and Interac. Data Anal.*, pp. 69–74, 2017.

[69] S. Ingram and T. Munzner. Dimensionality reduction for documents with nearest neighbor queries. *Neurocomputing*, 150, 2015.

[70] S. Ingram, T. Munzner, and M. Olano. Glimmer: Multilevel mds on the gpu. *IEEE Trans. Vis. Comp. Graph.*, 15(2):249–261, 2009.

[71] D. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. ipca: An interactive system for pca-based visual analytics. *Comp. Graph. Forum.*, 28(3):767–774, 2009.

[72] R. Jin, S. Wang, and Z.-H. Zhou. Learning a distance metric from multi-instance multi-label data. In *IEEE Conf. Comp. Vis. Patt. Recog.*, pp. 896–902, 2009.

[73] P. Joia, D. Coimbra, J. Cuminato, F. Paulovich, and L. Nonato. Local affine multidimensional projection. *IEEE Trans. Vis. Comp. Graph.*, 17:2563–2571, 2011.

[74] P. Joia, F. Petronetto, and L. G. Nonato. Uncovering representative groups in multidimensional projections. *Comp. Graph. Forum.*, 34(3):281–290, 2015.

[75] S. Kaski, J. Nikkilä, M. Oja, J. Venna, P. Törönen, and E. Castrén. Trustworthiness and metrics in visualizing similarity of gene expression. *Bioinformatics*, 4(1):1, 2003.

[76] D. A. Keim, F. Rossi, T. Seidl, M. Verleysen, and S. Wrobel. Information visualization, visual data mining and machine learning. *Dagstuhl Seminar*, 12081:79, 2012.

[77] H. Kim, J. Choo, C. Reddy, and H. Park. Doubly supervised embedding based on class labels and intrinsic clusters for high-dimensional data visualization. *Neurocomputing*, 150:570–582, 2015.

[78] J. Krause, A. Dasgupta, J. Fekete, and E. Bertini. Seekaview: An intelligent dimensionality reduction strategy for navigating high-dimensional data spaces. In *LDAV*, pp. 11–19, 2016.

[79] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[80] M. Kubovy. The perceptual organization of dot lattices. *Psychonomic Bulletin & Review*, 1(2):182–190, 1994.

[81] K. Lagus, T. Honkela, S. Kaski, and T. Kohonen. Self-organizing maps of document collections: A new approach to interactive exploration. In *KDD*, vol. 96, pp. 238–243, 1996.

[82] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. ivisclustering: An interactive visual document clustering via topic modeling. *Comp. Graph. Forum.*, 31(3pt3):1155–1164, 2012.

[83] J. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[84] J. Lee and M. Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7):1431–1443, 2009.

[85] J. Lee and M. Verleysen. Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants. *Procedia Comp. Sci.*, 4:538–547, 2011.

[86] J. Lee and M. Verleysen. Two key properties of dimensionality reduction methods. In *Symp. Comp. Intel. and Data Mining*, pp. 163–170, 2014.

[87] D. Lehmann and H. Theisel. Orthographic star coordinates. *IEEE Trans. Vis. Comp. Graph.*, 19(12):2615–2624, 2013.

[88] D. Lehmann and H. Theisel. General projective maps for multidimensional data projection. *Comp. Graph. Forum.*, 35(2):443–453, 2016.

[89] S. Lespinats and M. Aupetit. Checkviz: Sanity check and topological clues for linear and non-linear mappings. *Comp. Graph. Forum.*, 30(1):113–125, 2011.

[90] S. Lespinats, M. Aupetit, and A. Meyer-Baese. Classimap: A new dimension reduction technique for exploratory data analysis of labeled data. *Int. J. Patt. Recog. Artif. Intell.*, 29(6):1551008–1–1551008–45, 2015.

[91] J. Lewis, L. van der Maaten, and V. de Sa. A behavioral investigation of dimensionality reduction. In *CogSci*, pp. 671–676, 2012.

[92] D. Lin. An information-theoretic definition of similarity. In *ICML*, vol. 98, pp. 296–304, 1998.

[93] S. Liu, B. Wang, P.-T. Bremer, and V. Pascucci. Distortion-guided structure-driven interactive exploration of high-dimensional data. *Comp. Graph. Forum.*, 33(3):101–110, 2014.

[94] J. Lötsch and A. Ultsch. A machine-learned knowledge discovery method for associating complex phenotypes with complex genotypes. application to pain. *J. Biom. Info.*, 46(5):921 – 928, 2013.

[95] G. Mamani, F. Fatore, L. G. Nonato, and F. Paulovich. User-driven feature space transformation. *Comp. Graph. Forum.*, 32(3pt3):291–299, 2013.

[96] D. Marghescu. Evaluating the effectiveness of projection techniques in visual data mining. In *IASTED Int. Conf. Vis. Imag. Image Proc.*, pp. 186–193, 2006.

[97] R. Martins, D. Coimbra, R. Minghim, and A. Telea. Visual analysis of dimensionality reduction quality for parameterized projections. *Comp. & Graph.*, 41:26–42, 2014.

[98] A. Mayorga and M. Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Trans. Vis. Comp. Graph.*, 19(9):1526–1538, 2013.

[99] D. Meng, Y. Leung, and Z. Xu. A new quality assessment criterion for nonlinear dimensionality reduction. *Neurocomputing*, 74(6):941–948, 2011.

[100] B. Mokbel, W. Lueks, A. Gisbrecht, and B. Hammer. Visualizing the quality of dimensionality reduction. *Neurocomputing*, 112:109–123, 2013.

[101] A. Morrison, G. Ross, and M. Chalmers. A hybrid layout algorithm for sub-quadratic multidimensional scaling. In *Symp. Info. Vis.*, pp. 152–158,

2002.

[102] R. Motta, R. Minghim, A. Lopes, and M. Oliveira. Graph-based measures to assist user assessment of multidimensional projections. *Neurocomputing*, 150:583–598, 2015.

[103] T. Munzner. *Visualization Analysis and Design*. A.K. Peters visualization series. A.K. Peters, 2014.

[104] D. Niu, J. Dy, and M. Jordan. Dimensionality reduction for spectral clustering. In *Int. Conf. Artif. Intel. Stat.*, pp. 552–560, 2011.

[105] A. Nocaj and U. Brandes. Organizing search results with a reference map. *IEEE Trans. Vis. Comp. Graph.*, 18(12):2546–2555, 2012.

[106] H. Osipyan, M. Kruliš, and S. Marchand-Maillet. A Survey of CUDA-based Multidimensional Scaling on GPU Architecture. In C. Schulz and D. Liew, eds., *Imper. College Comp. Student Workshop*, vol. 49, pp. 37–45, 2015.

[107] L. Pagliosa, P. Pagliosa, and L. G. Nonato. Understanding attribute variability in multidimensional projections. In *Sibgrapi*, pp. 1–8, 2016.

[108] P. Pagliosa, F. Paulovich, R. Minghim, H. Levkowitz, and L. G. Nonato. Projection inspector: Assessment and synthesis of multidimensional projections. *Neurocomputing*, 150:599–610, 2015.

[109] S. Palmer. *Vision Science - Photons to Phenomenology (chap. 8.1)*. The MIT Press, April 1999.

[110] S. Palmer and S. Guidi. Mapping the perceptual structure of rectangles through goodness-of-fit ratings. *Perception*, 40:1428 – 1446, 2011.

[111] V. Pan and Z. Chen. The complexity of the matrix eigenproblem. In *Symp. Theory of Comp.*, pp. 507–516, 1999.

[112] A. Pandey, J. Krause, C. Felix, J. Boy, and E. Bertini. Towards understanding human similarity perception in the analysis of large sets of scatter plots. In *Conf. on Human Fact. in Comp. Sys.*, pp. 3659–3669, 2016.

[113] S. Park, S.-Y. Shin, and K.-B. Hwang. Cfmds: Cuda-based fast multidimensional scaling for genome-scale data. *Bioinformatics*, 13(17):1, 2012.

[114] F. Paulovich, C.Silva, and L. Nonato. User-centered multidimensional projection techniques. *Comp. Sci. Eng.*, 14(4):74–81, 2012.

[115] F. Paulovich, D. Eler, J. Poco, C. Botha, R. Minghim, and L. Nonato. Piecewise laplacian-based projection for interactive data exploration and organization. *Comp. Graph. Forum.*, 30(3):1091–1100, 2011.

[116] F. Paulovich and R. Minghim. Hipp: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Trans. Vis. Comp. Graph.*, 14(6):1229–1236, 2008.

[117] F. Paulovich, L. Nonato, R. Minghim, and H. Levkowitz. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Trans. Vis. Comp. Graph.*, 14(3):564 –575, 2008.

[118] F. Paulovich, C. Silva, and L. Nonato. Two-phase mapping for projecting massive data sets. *IEEE Trans. Vis. Comp. Graph.*, 16(6):1281–1290, 2010.

[119] F. Paulovich, F. Toledo, G. Telles, R. Minghim, and L. G. Nonato. Semantic wordification of document collections. *Comp. Graph. Forum.*, 31(3pt3):1145–1153, 2012.

[120] E. Pekalska, D. de Ridder, R. Duin, and M. Kraaijveld. A new method of generalizing sammon mapping with application to algorithm speed-up. In *Ann. Conf. Adv. Sch. Comp. Imag.*, pp. 221–228, 1999.

[121] N. Pezzotti, T. Höllt, B. Lelieveldt, E. Eisemann, and A. Vilanova. Hierarchical stochastic neighbor embedding. *Comp. Graph. Forum.*, 35(3):21–30, 2016.

[122] N. Pezzotti, B. Lelieveldt, L. van der Maaten, T. Hollt, E. Eisemann, and A. Vilanova. Approximated and user steerable tsne for progressive visual analytics. *IEEE Trans. Vis. Comp. Graph.*, 23(7):1739–1752, 2017.

[123] P. Rauber, S. Fadel, A. Falcao, and A. Telea. Visualizing the hidden activity of artificial neural networks. *IEEE Trans. Vis. Comp. Graph.*, 23(1):101–110, 2017.

[124] R. Rensink and G. Baldridge. The perception of correlation in scatterplots. *Comp. Graph. Forum.*, 29(3):1203–1210, 2010.

[125] B. Rieck and H. Leitte. Persistent homology for the evaluation of dimensionality reduction schemes. *Comp. Graph. Forum.*, 34(3):431–440, 2015.

[126] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological cybernetics*, 61(4):241–254, 1989.

[127] G. Rosman, A. Bronstein, M. Bronstein, A. Sidi, and R. Kimmel. Fast multidimensional scaling using vector extrapolation. *SIAM J. Sci. Comp.*, 2, 2008.

[128] P. Rousset and C. Guinot. Distance between kohonen classes visualization tool to use som in data set analysis and representation. In *Bio-Inspired Applic. Connectionism: Int. Work-Conf. on Artif. Natural Neural Networks*, pp. 119–126. 2001.

[129] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[130] D. Sacha, A. Stoffel, F. Stoffel, B. Kwon, G. Ellis, and D. Keim. Knowledge generation model for visual analytics. *IEEE Trans. Vis. Comp. Graph.*, 20(12):1604–1613, 2014.

[131] D. Sacha, L. Zhang, M. Sedlmair, J. Lee, J. Peltonen, D. Weiskopf, S. North, and D. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. Vis. Comp. Graph.*, 1(33):241–251, 2016.

[132] J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. on Comp.*, 18(5):401–409, 1969.

[133] S. Santini and R. Jain. Similarity measures. *IEEE Trans.Pat. Anal. Mach. Intel.*, 21(9):871–883, 1999.

[134] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *Int. Conf. Artif. Neural Networks*, pp. 583–588, 1997.

[135] M. Sedlmair and M. Aupetit. Data-driven evaluation of visual quality measures. *Comp. Graph. Forum.*, 34(3):201–210, 2015.

[136] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory. A taxonomy of visual cluster separation factors. *Comp. Graph. Forum.*, 31(3pt4):1335–1344, 2012.

[137] C. Seifert, V. Sabol, and W. Kienreich. Stress maps: Analysing local phenomena in dimensionality reduction based visualisations. In *EuroVAST*, pp. 13–18, 2010.

[138] R. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962.

[139] A. Shirkhorshidi, S. Aghabozorgi, and T. Wah. A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS One*, 10(12):e0144059, 2015.

[140] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Symp. Visual Languages*, pp. 336–343, 1996.

[141] S. Siegel. *Nonparametric statistics for the behavioral sciences.* McGraw-hill, 1956.

[142] R. Silva, P. Rauber, R. Martins, R. Minghim, and A. Telea. Attribute-based visual explanation of multidimensional projections. In *EuroVA*, 2015.

[143] R. Silva, E. Vernier, P. Rauber, J. Comba, R. Minghim, and A. Telea. Metric evolution maps: Multidimensional attribute-driven exploration of software repositories. In *Vis. Mod. & Vis.*, 2016.

[144] V. Silva and J. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Adv. in Neural Info. Proc. Sys.*, pp. 705–712, 2002.

[145] M. Sips, B. Neubert, J. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Comp. Graph. Forum.*, 28(3):831–838, 2009.

[146] A. Skupin. A cartographic approach to visualizing conference abstracts. *Comp. Graph. Appl.*, 22(1):50–58, 2002.

[147] L. Song, A. Gretton, K. Borgwardt, and A. Smola. Colored maximum variance unfolding. In *Adv. Neural Info. Proc. Sys.*, pp. 1385–1392, 2007.

[148] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Trans. Vis. Comp. Graph.*, 22(1):629–638, 2016.

[149] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. *Comp. Graph. Forum.*, 33(3):401–410, 2014.

[150] W. Tai and C. Hsu. A growing mixed self-organizing map. In *Int. Conf. Natural Comp.*, vol. 2, pp. 986–990, 2010.

[151] R. Tamassia. *Handbook of graph drawing and visualization*. CRC press, 2013.

[152] E. Tejada, R. Minghim, and L. G. Nonato. On improved projection techniques to support visual exploration of multi-dimensional data sets. *Info. Vis.*, 2(4):218–231, 2003.

[153] J. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[154] M. Tenenhaus and F. Young. An analysis and synthesis of multiple correspondence analysis, optimal scaling, dual scaling, homogeneity analysis and other methods for quantifying categorical multivariate data.

Psychometrika, 50(1):91–119, 1985.
[155] W. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.
[156] U.S. Geological Survey. An album of map projections. *Professional Paper 1453*, 2009.
[157] L. van der Maaten. Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res.*, 15(1):3221–3245, 2014.
[158] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *J. Mach. Learn. Res.*, 9:2579–2605, 2008.
[159] L. van der Maaten, E. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review. *J. Mach. Learn. Res.*, 10(1-41):66–71, 2009.
[160] J. J. van Wijk. Unfolding the earth: Myriahedral projections. *The Cartographic Journal*, 45(1):32–42, 2008.
[161] J. Venna and S. Kaski. Local multidimensional scaling. *Neural Networks*, 19(6):889–899, 2006.
[162] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J. Mach. Learn. Res.*, 11:451–490, 2010.
[163] J. Vesanto. Som-based data visualization methods. *Intell. Data Anal.*, 3(2):111–126, 1999.
[164] J. Wang and K. Mueller. The visual causality analyst: An interactive interface for causal reasoning. *IEEE Trans. Vis. Comp. Graph.*, 22(1):230–239, 2016.
[165] Y. Wang, K. Feng, X. Chu, J. Zhang, C.-W. Fu, M. Sedlmair, X. Yu, and B. Chen. A perception-driven approach to supervised dimensionality reduction for visualization. *IEEE Trans. Vis. Comp. Graph.*, 2017.
[166] Y. Wang and K. L. Ma. Revealing the fog-of-war: A visualization-directed, uncertainty-aware approach for exploring high-dimensional data. In *IEEE Int. Conf. on Big Data*, pp. 629–638, 2015.
[167] C. Ware. *Information Visualization: Perception for Design*. Elsevier, 2004.
[168] M. Wattenberg, F. Viégas, and I. Johnson. How to use t-sne effectively. Technical report, Distill, 2016.
[169] K. Weinberger and L. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *AAAI*, vol. 6, pp. 1683–1686, 2006.
[170] C. Williams. On a connection between kernel pca and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19, 2002.
[171] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *Symp. Info. Vis.*, pp. 57–64, 2004.
[172] J. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Symp. Info. Vis.*, pp. 51–58, 1995.
[173] A. Wismüller, M. Verleysen, M. Aupetit, and J. Lee. Recent advances in nonlinear dimensionality reduction, manifold and topological learning. In *ESANN*, 2010.
[174] Y. Wu, T. Provan, F. Wei, S. Liu, and K.-L. Ma. Semantic-preserving word clouds by seam carving. *Comp. Graph. Forum.*, 30(3):741–750, 2011.
[175] J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *J. Mach. Learn. Res.*, 6:483–502, 2005.
[176] X. Yuan, D. Ren, Z. Wang, and C. Guo. Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data. *IEEE Trans. Vis. Comp. Graph.*, 19(12):2625–2633, 2013.
[177] P. Zhang, Y. Ren, and B. Zhang. A new embedding quality assessment method for manifold learning. *Neurocomputing*, 97:251–266, 2012.
[178] S. Zhang. Enhanced supervised locally linear embedding. *Patt. Recogn. Lett.*, 30:1208–1218, 2009.
[179] T. Zhang, J. Yang, D. Zhao, and X. Ge. Linear local tangent space alignment and application to face recognition. *Neurocomputing*, 70(7):1547–1553, 2007.
[180] Z.-y. Zhang and H.-y. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *J. Shanghai Univ.*, 8(4):406–424, 2004.
[181] D. Zhao, Z. Lin, and X. Tang. Laplacian pca and its applications. In *IEEE ICCV*, pp. 1–8, 2007.
[182] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma. Stable principal component pursuit. In *Int. Symp. Inf. Theory*, pp. 1518–1522, 2010.

Biography of Authors

Dr. Luis Gustavo Nonato Luis Gustavo Nonato received the PhD degree in Applied Mathematics from the Pontificia Universidade Católica do Rio de Janeiro, Rio de Janeiro - Brazil, in 1998. His research interests include visualization, visual analytics, geometric computing, and data science. Nonato is full professor at the Institute of Mathematical and Computer Sciences - University of São Paulo, São Carlos, Brazil, and he is currently a visiting professor at the Center for Data Science - New York University, New York, USA. From 2008 to 2010 Nonato was a visiting scholar at the Scientific Computing and Imaging Institute - University of Utah, Salt Lake City, USA. Besides having served in several program committees, including IEEE SciVis, IEEE InfoVis, and EuroVis, he was associate editor of Computer Graphics Forum and currently he is associate editor of IEEE TVCG. Nonato is also the editor of the SBMAC SpringerBriefs in Applied Mathematics and Computational Sciences.

Dr. Hab. Michaël Aupetit was born in Limoges, France, in 1975. He went through 2 years of Higher School Preparatory Classes (CPGE) in Mathematics and Technology in 1994-1995 and received the Computer Science Engineering degree specialized in Artificial Intelligence from Ecole pour les etudes et la Recherche en Informatique et Electronique, Nîmes, France, in 1998, the M.Sc. degree in Robotics and Micro-electronics from University of Montpellier 2, Montpellier, France, in 1998, and the Ph.D. degree in Industrial Engineering from Institut National Polytechnique de Grenoble (INPG), Grenoble, France, in 2001. In 2002, he joined the Detection and Geophysics Laboratory (LDG) from the French Nuclear Energy Commission - Direction of Military Applications (CEA-DAM), as a postdoc, and then as a permanent researcher in 2004, designing and applying Machine Learning and Data Analysis techniques to support monitoring of seismic events. He moved to the Data Analysis and Intelligent Systems Laboratory (LADIS) from the French Nuclear Energy Commission - Technologies (CEA-Tech), in 2008 as a Senior Expert Researcher, designing and applying Machine Learning and Visual Analytics to solve problems from industrial partners. In 2012, he received the Accreditation to Supervise Research in Computer Science from the University of Paris-Sud 11 (Paris-Saclay Campus) for his collaborative research work on topological approaches for data analytics and decision support. His current research interests include visual analytics, dimension reduction, clustering, exploratory data analysis, topological data analysis, machine learning, and more generally how machines can efficiently support human decisions through interactive visualization. Since 2014, he worked as a Research Scientist at the Qatar Computing Research Institute in Doha, Qatar, to tackle national challenges related to health and cybersecurity.