Visualizing nD Point Clouds as Topological Landscape Profiles to Guide Local Data Analysis

Patrick Oesterling, *Student Member, IEEE Computer Society,* Christian Heine, *Member, IEEE Computer Society,* Gunther H. Weber, *Member, IEEE Computer Society,* and Gerik Scheuermann, *Member, IEEE*

Abstract—Analyzing high-dimensional point clouds is a classical challenge in visual analytics. Traditional techniques, such as projections or axis-based techniques, suffer from projection artifacts, occlusion, and visual complexity. We propose to split data analysis into two parts to address these shortcomings. First, a structural overview phase abstracts data by its density distribution. This phase performs topological analysis to support accurate and non-overlapping presentation of the high-dimensional cluster structure as a topological landscape profile. Utilizing a landscape metaphor, it presents clusters and their nesting as hills whose height, width, and shape reflect cluster coherence, size, and stability, respectively. A second local analysis phase utilizes this global structural knowledge to select individual clusters or point sets for further, localized data analysis. Focusing on structural entities significantly reduces visual clutter in established geometric visualizations and permits a clearer, more thorough data analysis. This analysis complements the global topological perspective and enables the user to study subspaces or geometric properties, such as shape.

Index Terms—Point clouds, high-dimensional data, cluster analysis, dimension reduction, scalar topology, and visual metaphors.

1 Introduction

Analyzing real-world phenomena often requires the identification of groups among observations and judging group cohesion and separability. Observations are usually encoded as high-dimensional points (feature vectors). Popular applications include data mining, analyzing gene expression data, classifying images, or understanding document collections. One example is the analysis of a dataset describing the composition of olive oils with a feature vector consisting of percentages of eight fatty acids. In this example, one is interested if these oils form clusters based on their combination of fatty acids, and whether these clusters correspond, e.g., to geographic growing regions. Another example is the classification of newspaper articles where a user is often interested in finding “theme“ groups (such as sports, or politics) and their relation to each other.

Analysis via clustering methods, and visualizing points with techniques that rely on the human visual system to identify structure only in the final visual representation are common approaches to identify structure in such datasets. Examples include axis-based techniques, such as parallel coordinate plots (PCP) [1], scatter plots [2], and projections, such as principal component analysis (PCA) [3]. However, these techniques are restricted by underlying assumptions about the data’s structure. Since each point is represented by at least one pixel, they also suffer from occlusion (at the latest) when the size of the dataset exceeds that of the screen. Projective approaches, moreover, have a fundamental problem: although they rely on visual extraction of structure (conveyed by distance), they can not ensure distance preservation for data with more than two dimensions. Occlusion and illusions are thus inevitable for high-dimensional data.

Because a visualization cannot preserve both structure and geometric details at the same time, we propose to analyze them separately. At first, we neglect geometric properties (such as distance and shape) to ensure an adequate, occlusion-free display of a dataset’s structure. Afterwards, we utilize this structural perspective to select single clusters for further (geometric) analysis in multiple linked-views [4]. Scalability issues are thus solved by the assumption that we will end up with fewer features than data points, and that further analysis is applied to only a few features at one time.

Fua *et al.* [5] also used this general concept, terming it *structure-based brushing*, to navigate in hierarchical organized data. Their approach uses hierarchical cluster trees and permits brushing-and-linking of selected subtrees to highlight aggregated data in several linked visualizations. We improve this concept by providing additional information about feature relevance and by supporting more sophisticated selection in the structural view. Displaying cluster quality helps to identify interesting features before linking them to other views.

To show a structural perspective, we use the results of previous work on high-dimensional cluster visualization [6]. Fig. 1 provides an overview of the approach

- P. Oesterling, C. Heine, and G. Scheuermann are with the Computer Science Dept, University of Leipzig, PF 100920, D-04009, Germany, E-mail: {oesterling,scheuermann}@informatik.uni-leipzig.de.
- C. Heine is also with the Department of Computer Science at ETH Zürich, 8092 Zürich, Switzerland, E-mail: cheine@inf.ethz.ch.
- G.H. Weber is with the Computational Research Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA-94720 and the Institute for Data Analysis and Visualization, Department of Computer Science, University of California, Davis, 1 Shields Avenue, CA-95626, E-mail: ghweber@lbl.gov.

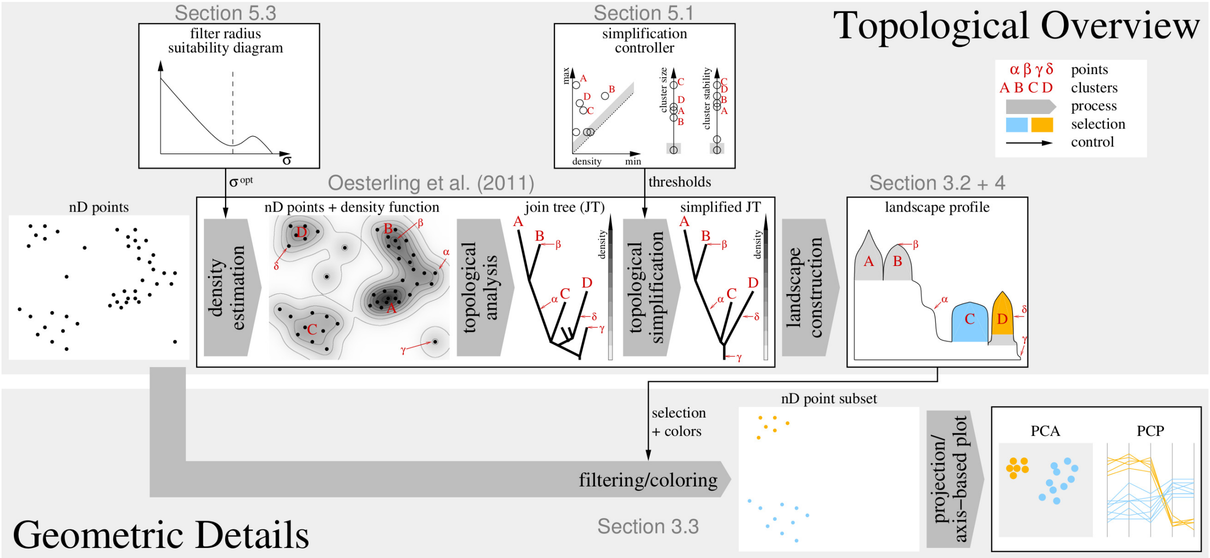


Fig. 1. Framework for exploring high-dimensional datasets. We obtain a structural perspective onto the data by analyzing the input point cloud's density function topologically. Cluster structure and cluster properties are encoded as a *join-tree*. The topological analysis and the join tree's complexity depend on parameters for whose selection we present interactive controllers. The join tree is finally visualized as a landscape profile. Hills, their nesting, size, and shape adequately reflect the high-dimensional clustering. We also present specific selection mechanisms to isolate structural features for linking to PCA- and PCP-views for local analysis.

and indicates which of its parts this paper extends. We approximate the input data's density function and identify clusters as regions of high density. Quantitative properties of dense regions, such as distinctness or the number of points, then describe cluster quality and their relevance. As we cannot visualize the high-dimensional density function occlusion-free in two dimensions, we focus only on its structure. To this end, we analyze the density function's topology by considering at which densities regions merge with other regions. The nesting of regions, including quantitative properties for each of them, are then encoded in a *join tree* [7], on which we finally base our structural perspective.

We improve our previous work by presenting a novel topological landscape profile representation of the join tree. Clusters, including their hierarchy and quality measures, are represented occlusion-free as hills of different size in the landscape profile (cf. Fig. 2). We add an additional cluster property, cluster stability, to further improve perception of cluster relevance in the structural view. This landscape representation permits extraction of a dataset's clustering structure and supports examining each cluster's size, compactness, and variance. Compared to the topological landscape used in [6], proper feature identification and comparison is simplified because all features are simultaneously visible without changing the view.

Since local geometric properties are also important to explain *why* clusters have sub-clusters, or in which dimensions clusters differ, we complement the global analysis with linked views for further local analysis. We augment the landscape profile with the input data and present specific selection mechanisms to enable brushing-and-linking to PCA- and PCP-views.

Our framework depends on several parameters that affect the visualization in terms of accuracy and visual clarity. As these parameters require sophisticated adjustment, we present interactive widgets to help users to choose these parameters appropriately.

2 Related Work

Projective methods aim to exploit the ability of the human visual system to group and separate points based on spatial closeness. Examples include *principal component analysis* (PCA) [3], *Self-organizing maps* (SOM) [8], or *Multidimensional scaling* (MDS) [9]. These approaches are restricted by underlying assumptions about the data's structure, e.g. linearity for PCA, dimensionality of the manifold for SOM, and neglect for the curse of dimensionality in MDS. This often results in distortions and overlaps, causing illusionary artifacts and clutter.

Structural overview + local details. Separating global structural analysis from local geometric details is in accordance with the visual information-seeking mantra: "overview first, zoom and filter, then details-on-demand", as proposed by Shneiderman *et al.* [10]. The utilization of several interactive plots used for brushing-and-linking of features to other views relates to the concept of multiple coordinated views. Roberts *et al.* [4] provides an overview of this concept.

Using these concepts for visual cluster analysis, Fua *et al.* [5] introduced *structure-based brushes* to navigate through hierarchical cluster trees. They use a special

2D brushing tool to permit individual selection of subtrees at different levels-of-detail. In [11] they applied this technique to PCPs by aggregating data as bands of varying translucency. Yang *et al.* [12] extended this work to support other traditional visualization methods, such as star glyphs [13] or scatterplot matrices [14].

We extend the concept of structure-based brushes by adding several information about cluster quality and relevance to the structural perspective. This information facilitates easier feature identification prior to linking to other views. We also note that our structural view does not consider a hierarchical clustering, but a cluster hierarchy for *one* level-of-detail. While in [5] level-of-detail determination and feature selection are combined in one brushing tool, we consider them separately to support more sophisticated feature selection.

Johannson *et al.* [15] proposed the use of pre-clustering to present inherent data structure via high-precision textures and different transfer functions. Novotny *et al.* [16] used clustering on a binned data representation to combine outliers, trends, and focused data items in an aggregated PCP. Other techniques to present hierarchical structure include tree-layouts [17], dendrograms [18], or Icicle Plots [19]. However, these methods present little information beyond hierarchy and thus provide only a one-sided global perspective.

Density-based clustering + kernel width. We use a density-based approach, akin to scalespace analysis [20], for our structural perspective. Such techniques, including DENCLUE [21], OPTICS [22], or DBSCAN [23], assume one density peak per cluster and low density between clusters. Density-based clustering is designed to find clusters of arbitrary shape, as long as dense regions are separated by regions of low density. However, this approach is subject to an important parameter: the kernel window width σ. This value has crucial influence on the clustering. A value too large can merge separate clusters, and a value too small can force clusters to split. Some heuristics determine σ by analyzing the k-nearest neighbors [23], or by choosing σ such that a bigger modification does not change the number of identified density maxima [21]. However, this property often depends on the data itself and does not necessarily reflect that anisotropic clusters can feature several density maxima. To help users in finding this parameter in a descriptive way, we present widgets that rely on topological concepts.

Topology-based visualization. Since we cannot visualize the high-dimensional density function in 2D, we only consider (and visualize) its topological structure.

Using topological concepts to visualize high-dimensional scalar functions has been an actively researched field for some years. Weber *et al.* [24] introduced *topological landscapes*, a 3D terrain metaphor that has the same topology (of the height values) as an arbitrary dimensional input scalar function. Maxima and minima of the function show up as hills and sinks in the terrain, thus accurately reflecting high-dimensional information in 3D. Subsequent work aimed at eliminating limitations in the initial implementation (mainly regarding accuracy and usability) of this metaphor. Harvey *et al.* [25] used a *tree map* [26] construction scheme to improve the accuracy of mapping a measure to feature area in the terrain. Oesterling *et al.* [27] utilized a flattened version that conveys structure by color to avoid occlusion of features. However, as colors are not always easily distinguishable we illustrate structure by height values in our landscape profile. This representation allows us to use color as an additional information channel.

Takahashi *et al.* [28] adopted the well-known *ISOMAP* [29] algorithm and proposed a 3D arrangement of the input positions that reflects the topology as a tree-like structure. Gerber *et al.* [30] proposed a method that combines topological and geometric techniques to provide interactive visualizations of discretely sampled high-dimensional scalar fields. They use an approximate Morse-Smale complex embedded in 2D space.

Note that these techniques are naturally three-dimensional or convey structure by color, because they were designed to describe more complex topology. Since some topological events are irrelevant for density-based clustering, we can illustrate the same structure adequately in two dimensions.

Visual aids for parameter choice. Our approach depends on parameters for which we present widgets to enable convenient determination. We base our controller design on the concept of *Scented Widgets* [31], user interface components enhanced with embedded visualizations to quickly judge interesting thresholds. The idea to determine the parameters interactively is also in line with *Dynamic Queries*, as described by Ahlberg *et al.* [32].

3 Analysis and Framework Design

In earlier work [6], we developed an approach extending the concepts of density-based clustering. We proposed to use the topology of a high-dimensional point cloud's density function to obtain a clustering hierarchy. Fig. 1 illustrates this approach and also indicates which of its components this paper extends.

As input we consider a set of high dimensional points $P = \{p_1, \ldots, p_k\} \subseteq \mathbb{R}^n$, from which we first construct a neighborhood graph ([33]). We then sample the density function $dens : P \to \mathbb{R}$ at the input points and at the center of the neighborhood graph's edges using simple Gaussian kernel density estimates, subject to the *filter radius* σ. In [6] we presented heuristics for the determination of this parameter, but in this paper we suggest selecting the parameter in an interactive widget.

The density function is then studied topologically. By mapping density to height, the density function can be conceptually thought of as a high-dimensional height field. This landscape is then flooded with water and successively drained, whereby the connectivity of land mass is studied. The *join tree* encodes for varying height

h the joining of regions with minimum height h. As h is continuously decreased from the maximum, regions are created at local maxima, grow, and join at special values called *saddles*. For $h = 0$ there is only one region left, which encompasses the whole domain. This process is illustrated in Fig. 1 using contours. Data points map to exactly one edge in the join tree based on the height for which they become part of a region. The data points are needed to determine quantitative properties of each region, and to illustrate them in our landscape profile. We therefore store a sorted list of point-height pairs for each edge. Carr *et al.* [7] describe an efficient algorithm for computing the join tree.

Each subtree of the join tree represents a cluster $C \subseteq P$, and the tree encodes their nesting. We note that in contrast to hierarchical cluster trees, the join tree provides a cluster hierarchy for *one* level of detail. Furthermore, each edge can be assigned a cluster quality measure. In prior work, we used two topology-driven quality measures: persistence and cluster size. In this paper we introduce cluster stability as third quality measure (Section 3.1).

Topological simplification, based on cluster quality thresholds, removes noise in the data and leaves only prominent clusters. While in earlier work we used heuristically chosen values, in this work we present widgets for threshold specification. Simplification does not remove data points, but moves them to parent clusters.

This simplified tree is shown in our previous approach using the topological landscapes of Weber *et al.* [24]. In the landscape, clusters show up as hills, and we placed data points as small spheres at suitable locations in the landscape to enable detailed exploration. As our method only uses a part of the density function's topological features, we propose to show it as a topological landscape profile, enabling simpler overview and interaction. In this representation, the nesting of hills still reflects cluster hierarchy, and a hill's height shows the maximum density of its corresponding cluster. The profile makes it possible to use a hill's width to show precisely the number of data points belonging to it and a hill's shape to show the density distribution of points. It also enables brushing-and-linking of data subsets for further inspection via traditional methods.

3.1 Quality Measures and Cluster Relevance

Sub-trees of the join tree correspond to (sub-)clusters in the input data. To enable cluster comparison we use three cluster quality measures:

A cluster's size $size(C) = |C|$ is the number of points it contains and therefore also the total of the corresponding tree edges' associated point lists' sizes. As a cluster's persistence [34] $pers(C) = \max_{p \in C} dens(p) - \min_{p \in C} dens(p)$ we denote the difference of the sub-tree's minimum and maximum density. For separated clusters, the minimum density is zero. Borrowing from the idea of hyper-volume, we define a cluster's stability $stab(C) = \sum_{p \in C} dens(p)$ as the sum of the contained points' densities. Conceptually, this measure represents the amount of energy required to erode the cluster, and will thus be reflected by a hill's area in our landscape profile.

With these topology-driven quality measures, we are able to determine cluster significance and a cluster's relevance compared to other clusters. In contrast to distance or shape it is also possible to preserve these measure without loss in a topological landscape profile.

High density usually reflects data coherence. If feature vectors are very close, they heavily contribute to their mutual density. Therefore, a cluster's density reveals where points are very similar in feature space. Persistence, on the other hand, can be used to judge the distinctiveness of a cluster regarding its surrounding regions. By contrasting a region's density, or persistence to its number of points, we can also derive information about cluster compactness or variance. While a few points of high density must be very compact, many points without significant density will be scattered. Furthermore, the density distribution within a cluster also provides information about coherence. For a cluster to be relevant, we expect the points to accumulate at high density, rather than being spread on a wider density range. This behavior is reflected by the stability, which is maximal if all points in a cluster have the same density.

These observations about compactness, variance, and point distribution are very relevant in many application domains. For example, they indicate how well documents are arranged around their *topical center*, i.e. how similar their content is. For image data, where clusters correspond to images with similar content, the compactness tells us how well different pictures match the mean-image of a particular motif or scenery.

3.2 Visual Representation of Global Structure

A clear visual description of the high-dimensional clustering requires an adequate visual representation of the abstract join tree. One obvious solution would be a tree-layout in the plane. However, to support comparison of several edge properties of possibly large trees, this layout would have to adhere to several conventions that might result in unfeasible overviews for our second analysis phase. Weber *et al.* proposed the *topological landscapes metaphor* [24], a terrain visualization that has the same topology (of the height values) as an input tree. In this metaphor, structure is conveyed by (sub-)hill relationships and two edge properties can be reflected by a hill's height and footprint. Since this metaphor was designed to represent more complex topological relationships than required for clustering, the terrain is three-dimensional (3D). In 3D, however, proper feature comparison is impeded by occlusion and perspective distortion. A 3D landscape also prohibits comparison of a hill's height and extent at the same time. Occlusions are only removed in a top view, where the height of hills is invisible. Furthermore, a hill's footprint was only approximately correct in the original approach, and it could take any possible shape; which makes direct comparison infeasible. Both drawbacks were eliminated by Harvey *et al.* [25] who used tree-maps for their landscape construction. However, although tree-maps always ensure correctly sized and evenly shaped hills, rectangular features can still be hard to compare for very different aspect ratios [35].

To permit an intuitive global overview, we base our visual representation on the topological landscapes metaphor. Since clustering results in a topologically much simpler tree structure, we can adapt this metaphor to two dimensions (2D). We propose a topological landscape profile for natural illustration of the clustering (represented by the join tree) as a set of hills that accurately describes hierarchy and edge properties by sub-hill relationships and a hill's height, width, and area, respectively. Due to orthographic projection, each cluster's properties are always correct and simultaneously visible. This property makes feature comparison easy and feasible without adjusting the viewing direction.

Fig. 4 shows a landscape profile for the image segmentation dataset from the UCI machine learning repository [36]. The dataset describes a fragmentation of seven outdoor images into 2,310 3x3 pixel regions. Each region is characterized by 19 attributes (including position, line densities, edges, and color values) and manually classified into one of the following types: brickface, sky, foliage, cement, window, path, and grass. Since height values represent density, sub-cluster relationships are reflected by valleys at non-zero height. A valley at zero-density reflects complete separation. A cluster's persistence, size, and stability are precisely reflected by its hill's respective height, width, and area.

For each height value, a hill's width reflects the number of points in the cluster having at least this density. This representation of a dense region's point distribution as a hill's shape can serve as a cluster quality measure and provides insights about data coherence or separation. For example, points of a well-separated cluster feature a significantly higher density than the cluster's merge density with another cluster (which is zero in this case). This concentration of high density makes the hill look rectangular-shaped, i.e., very stable. This observation also implies that triangular-, or peak-shaped hills represent not so compact and isolated clusters. Here, the densities of the points are distributed between merge-level and cluster maximum, which makes the cluster unstable. Finally, if a single hill features plateaus at different height levels, we know that this cluster consists of several groups that are not yet separable (suggesting that the currently chosen filter radius is too large).

In the second analysis phase, we shift attention from the global clustering to local data analysis. For this purpose, we augment the landscape profile with histograms to summarize the input point distribution based on the points' density and cluster affiliation. Possibly available classification information is used to extend the histograms to stacked barcharts, one bar and color per class. With this representation, as shown in Fig. 5, we can i) quickly determine whether classes correspond to clusters by analyzing the colors of the histograms on the hills, ii) facilitate labeling and semantic zoom based on metainformation, and iii) use brushing-and-linking to link brushed subsets to other views for local analysis.

We note that (Euclidean) distance, either between hills or histograms, has not always a meaning in the profile's topological perspective. If hills share valleys at zero-height, we just consider the clusters to be separated, no matter how far away they are from each other in the original domain. Only if hills are connected by a valley at non-zero height, we can derive closeness due to region-overlap Furthermore, points summarized by histograms are not necessarily close to each other. They only share the same density within the cluster, thus likely being arranged around the cluster's density maximum. We accept these restrictions as (inter-cluster) distance preservation is not necessarily needed to describe a clustering, and because it is only disregarding geometric properties that permits overlap-free visualization in the first place.

3.3 Interaction and Local Analysis

The topology-based global overview via the landscape profile already permits convenient and primarily overlap-free insights regarding the clustering's hierarchy and several quantitative properties. However, using only this global perspective on the data, we cannot tell *why* clusters have sub-clusters, or why points of the same class are not in one single cluster. A nested, hilly structure in the landscape profile only tells us that *globally* there is a dense region with several local density maxima. If one is interested in properties beyond a global perspective, we need to determine in which dimensions or subspaces similar data entities differ *locally*.

Utilizing global structural knowledge for local data inspection has two main advantages: First, since the overview minimizes the topological rather than the geometrical error, we have a reliable and convenient way to select all occurring clusters. Projections or axis-based approaches cannot ensure an appropriate selection because they suffer from occlusion and projection artifacts. If clusters are overplotted, we neither know their real structure nor could we appropriately select a single cluster for further analysis. The second advantage is that we can greatly reduce visual complexity of other visualization techniques if we focus on single structural features rather than on the whole dataset.

3.3.1 Feature selection in the Landscape Profile

We consider either a whole hill (i.e., a cluster), or a part of a histogram as a *feature*. Because selection in a particular hill's histogram is only meaningful for bars of different height or class, we ensure a bar-wise selection of the data points. We provide the following mechanisms:

1. **Whole clusters** can be selected by clicking on a hill in the landscape profile. After changing the color of the hill, linked visualization techniques subsequently use this color or the colors corresponding to the points' classes.
2. Selection of **arbitrary bars** is achieved by surrounding them with a selection rectangle or by individually clicking on them. Selections can be concatenated. Linked visualization techniques then use the colors corresponding to the classes.
3. To **distinguish points of one class** (color), subsequent selections can also be associated with a set of pre-defined, distinct colors. This association is especially helpful to select bars of the same class but at different heights. Linked visualization techniques use the colors of the individual selections.

3.3.2 Subspace Analysis with Axis-Based Methods

Axis-based techniques have great potential to analyze the homogeneity of subsets throughout many dimensions. However, due to their visual complexity and occlusion problems, the number of poly-lines to be visualized should be limited or important structure should at least be highlighted to avoid overloaded visualizations.

Since we visualize structure overlap-free with the landscape profile, we can use this global knowledge to focus on single clusters or point sets. This localization greatly reduces the number of poly-lines, and therefore the amount of visual overlapping (crossings). We can rely on an implicit correlation between the landscape profile and axis-based techniques: *Hills in the landscape profile correspond to poly-line-bundles in axis-based visualizations*. This correlation results from the fact that points of a global cluster necessarily have to share similar values throughout many dimensions.

To analyze why clusters consist of sub-clusters, or why points of the same class are not in the same cluster, we link selected features to a *PCP-view*. Note how selecting whole hills easily colors line bundles in axis-based visualizations, even if classification information is not available for the input data.

3.3.3 Geometric Properties Using Projections

Projections often do not preserve geometric properties of high-dimensional data sufficiently. Even for well-separated, low-dimensional clusterings a 2D projection likely contains some overlapping regions or inexact shapes if the clusters are oriented just differently enough.

For example, in PCA the greatest variance of multiple clusters certainly does not reflect the greatest variance of each individual cluster because all points are considered as a whole. That is, distances (and thus shapes) are represented less accurately if other points take part. In order to minimize the projection error, the optimization criterion should thus be applied to only a few points.

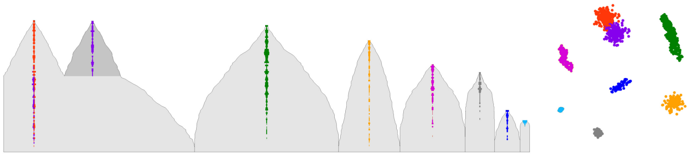


Fig. 2. Landscape profile (left) for an idealistic 2D clustering (right) with clusters of different compactness, size, and variance. Adjacent hills that share valleys at the same height can be sorted for easier comparison.

To select individual clusters reliably, we use the global knowledge that is adequately conveyed by the landscape profile. We only project points of whole hills or arbitrary features, thus maximizing the optimization criterion solely for these points. As a representative for projective methods, we implemented the well-known PCA because its underlying concept is generally accepted and very intuitive. Typically, other projective methods also reflect a single cluster's geometric properties more accurately if unselected points are omitted. We link selected features to the *PCA-view* and specify the projection error as the variance that is not explained by the first two principal components. Although this error can still be rather large for high-dimensional data, we keep it smaller by focusing on single structural entities.

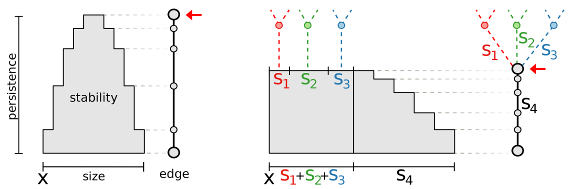
4 Implementation

4.1 Landscape Profile

The fundamental property of the landscape profile is to have the same topology (of the height values) like the density function's join tree. That is, each horizontal cut through the profile intersects as many hills as the same cut would intersect edges in the join tree. There is only little space for spatial optimizations in the profile without violating this precondition. One possible modification, however, is re-ordering a node's sub-trees, which enables sorting hills by relevance. Prior to generating the profile we sort each node's sub-trees by persistence to make similarly persistent hills appear next to each other. This gives the profile a global downward trend from left-to-right and enables easy feature comparison, especially for well-separated clusterings where valleys are all at zero-level (cf. Fig. 2). Sorting by cluster size or stability is also possible. It is even possible to sort hills by inter-cluster distance in the original domain. However, the additional knowledge to be expressed with only two neighbors is assumed to be rather insignificant for high-dimensional data. We therefore decided to use topology-driven relevance measures instead because they better reflect the topological principles of the profile itself and because they can be preserved without any loss in 2D.

4.1.1 Construction

To construct the landscape profile we use a simple recursive algorithm. Because each part of the landscape profile corresponds to an edge in the join tree, we just



(a) (b)

Fig. 3. (a) A leaf edge is mapped to a hill with a width and a height of that edge's associated number of points and persistence, respectively. The hill's shape is furthermore affected by the occurrences and densities of the data points associated with that edge. The area of the hill then reflects the edge's stability. (b) Non-leaf edges reserve space for child nodes and represent the edge to the right. s_1, s_2, s_3 represent whole sub-tree sizes. Red arrows indicate the currently processed node. In both cases, a binning approach can reduce the amount of produced geometry (at the cost of correct shape and stability).

Algorithm 1 Pseudo-code to construct the profile

Require: *root* node of the density function's join tree
Ensure: topological landscape profile

```
procedure PaintLandscapeProfile($xCoordinate$)
    PaintPart( $root$, $xCoordinate$)

procedure PaintPart( $node$, $x$)
    if HasParentEdge( $node$) then
        $edge \leftarrow$ GetParentEdge( $node$)
        if IsLeaf( $node$) then
            DrawHill( $edge$, $x$)                ▷ cf. Fig. 3a
        else
            DrawInnerPart( $edge$, $x$)           ▷ cf. Fig. 3b
    for $i \leftarrow 1$ to NumberChildNodes( $node$) do
        $childNode \leftarrow$ GetChildNode( $node$, $i$)
        PaintPart( $childNode$, $x$)
        $x \leftarrow x +$ SubTreeSize( $childNode$)
```

need to traverse the tree and consider each edge's (dense region's) persistence, number of points, and stability.

For simplicity, y-coordinates in the profile match the tree nodes' densities and we start with the root node at a certain x-coordinate. Algorithm 1 and Fig. 3 provide a detailed description of the construction process. A node's *parent edge* and *child edges* are its edges towards the tree's root node and the leaves, respectively. The root node has no parent edge, and the leaves have no child edges. Together with each edge we store a list of data points that make up the corresponding dense region.

We use a dual-color scheme to support visual extraction of hills and their hierarchy. Starting with one color for hills belonging to the root node, sub-hill relationships are subsequently emphasized by switching the colors for each hierarchy level. Note how this color scheme permits easy identification of separated clusters as equally colored hills sharing a valley at zero-level (cf. Fig. 4). To avoid visual confusion, we require both colors to be distinguishable from the colors used for the histograms. However, the user can still assign other colors to the hills if classification information is unavailable to color the histograms, or if data points in linked views should be highlighted based on the colors of their corresponding hills. We demonstrate hill coloring as part of the local data analysis in Section 6.2 and in the supplemental video material.

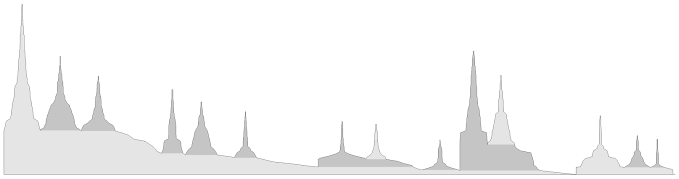


Fig. 4. Topological landscape profile of the image segmentation dataset. Hills correspond to dense regions. A hill's height, width and area reflects the corresponding cluster's persistence, size and stability, respectively.

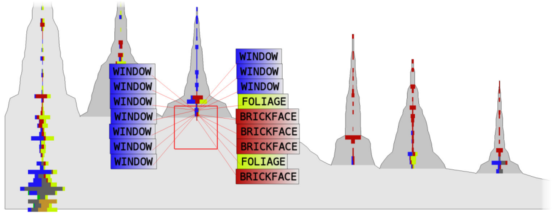


Fig. 5. Histograms are used to augment the profile with the input data, based on their density and cluster affiliation. Classification information is used for coloring and extension to stacked barcharts. The data can also be annotated with labels.

4.1.2 Data Point Representation

The landscape profile is augmented with (horizontal) histograms to display the input data distribution based on their affiliation to clusters. The width of a histogram bar is rescaled so that it reflects appropriately the number of its summarized points. A bar's height is subject to a granularity parameter that controls the smoothness of the distribution. If classification information is available we permit class-wise selection by extending the histograms to stacked barcharts, one bar per class.

For a clearly arranged visualization, histograms are placed centrally on the hills. For inner parts of sub-hill structures, however, we place them below the left-most hill. This placement is achieved by assigning the histogram of an inner edge the x-coordinate of the hill corresponding to the edge's first subtree's leftmost leaf edge.

To provide additional information, we label the points summarized by a histogram bar with the *excentric labeling* [37] approach. Using a movable focus area, labels of enclosed points are placed around the focus and they

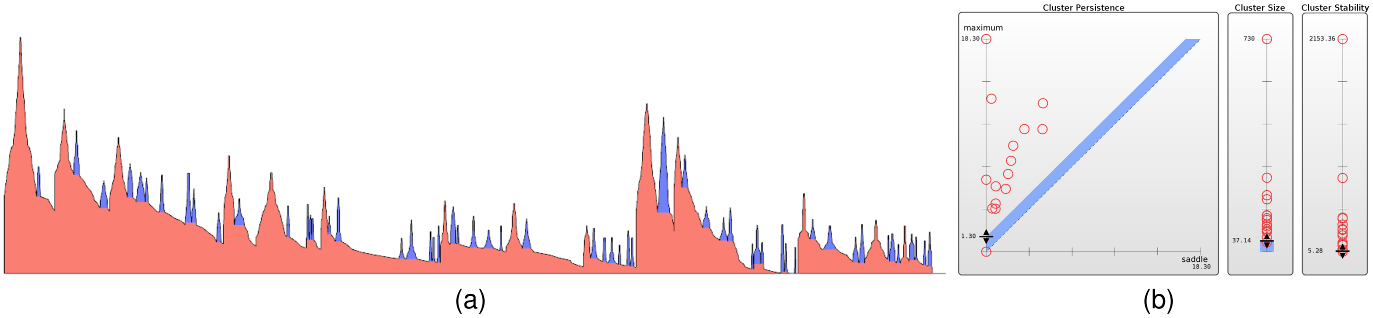


Fig. 6. (a) Small, thin, or little stable hills can be removed from the landscape profile to increase visual clarity. (b) Scented widgets in the simplification controller signify interesting thresholds to preserve only the most prominent features. The controller and the profile are linked to highlight in real-time which hills will remain after simplification.

are connected by a line. The label text can easily be changed to provide details-on-demand, e.g., more meta-information when the user zooms in. Fig. 5 shows some histograms for the image segmentation dataset. Here, colors reflect class affiliation and the labels specify the given classes. In general, a label's text, color, size and shape could be used to highlight different data aspects.

5 Parameter Widgets

The landscape profile can accurately present a high-dimensional clustering, including quantitative properties, overlap-free in 2D. However, both the underlying topological analysis [6] and the visual representation are each subject to one parameter that affects the profile in terms of correctness and visual clarity. These parameters are the filter radius σ for the density estimation, and a simplification threshold used to eliminate topological noise. Since these parameters require sophisticated adjustment, we present scented widgets [31] to support the lay user in a descriptive way.

5.1 Interactive Simplification Controller

Depending on the filter radius, a point cloud's density function usually contains small variations. They occur in noisy regions, at small point accumulations, or at outliers. This topological noise causes very small and thin hills in the landscape profile and thus disturbs its visual clarity. This behavior can be alleviated by peeling off *less relevant* leaf edges from the join tree prior to profile construction. For clustering it makes sense to assign the data points associated with removed edges to their parent edges. This way, a cluster maintains the relevance of simplified sub-clusters and all data points are preserved for representation in the profile. Depending on the user's preference, edges are removed if they do not feature enough persistence, number of points, or stability. Three controllers are provided for these measures.

Based on the idea of scented widgets [31], each controller highlights the join tree edges' distribution in terms of the property controlled by that widget. This allows the user to quickly judge interesting thresholds if some edges stand out. Because edge properties change during simplification, the controllers use a join tree segmentation called the *branch decomposition*: a multi-resolution representation as described by Pascucci *et al.* [38]. Similar to the simplification process, the branch decomposition is obtained by merging leaf edges with adjacent edges. The order in which leaf edges are simplified defines the hierarchy of branches and is based on a certain edge property. This principle was also used by Carr *et al.* [39] to simplify topological structures based on local geometric measures in 3D. Because branches already provide a multi-resolution view on the join tree, they make the controller's *scent* less susceptible to variations between two simplifications. They also indicate which thresholds are necessary to preserve only prominent features.

Fig. 6b shows the interactive simplification controller. It contains three slider widgets, one for each quality measure. The persistence threshold is adjusted with a *persistence diagram* [40], a 2D scatter plot mapping a branch's minimum and maximum density to an x- and y-coordinate, respectively. Since the persistence of a branch is the difference of these two values, topological noise shows up as points near the main diagonal (cf. Fig. 7d). A persistence diagram permits determination of relevant branches at different merge densities (horizontal axis) and it provides a quick way to determine the ratio of topological noise to interesting branches of high persistence. The latter correspond to points far away from the diagonal and represent dense regions surrounded by regions of low density. However, the diagram abstracts from branch hierarchy and branch affiliation to dense regions. Beyond persistence, it thus provides only a distorted idea of the clustering.

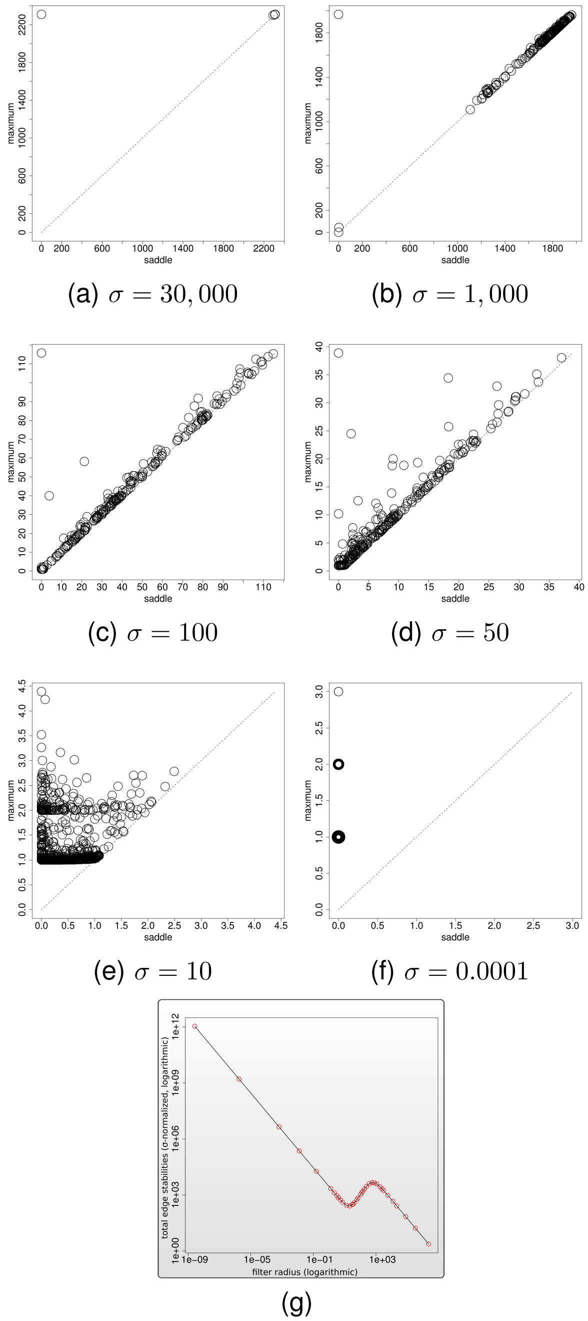
Thresholds are set by dragging sliders on the vertical axes. Parameter settings are also indicated as blue shaded regions in the widgets. When the user drags a slider, the join tree is simplified in real-time. Edges that do not satisfy all three thresholds are removed and the controllers are updated with their respective branch decompositions. The controllers are also linked to the landscape profile. For each change, hills that will be preserved are colored in red. Hills that will be removed are colored in blue (cf. Fig. 6a). When the user releases a slider, the landscape profile is reconstructed (here, resulting in the profile shown in Fig. 4). We refer to the supplemental video material for a demonstration of the simplification controller.

5.2 Filter Radius Setup with a Persistence Diagram

Setting the filter radius for the topological analysis, as described in [6], can be challenging. A filter radius too large will result in the whole point cloud appearing as one dense region, while a filter radius too small will result in each data point being considered a cluster of its own. Surely, the correct answer has to lie somewhere in-between and thus we need to provide the user with visual assistance to set up this parameter conveniently.

We analyzed the behavior of the density function's topology by observing persistence diagrams while varying the filter radius between the two extremes. We found a general behavior for all datasets we observed. In the following, we explain this behavior on the image segmentation dataset.

We start with a filter radius σ that is too large. The persistence diagram in Fig. 7a consists of the root branch as a single point in the upper left corner and a few branches of near-zero persistence, i.e. topological noise, in the upper right near the diagonal. Reducing the filter radius in Figs. 7b-c causes the small branches to start spreading along the diagonal while the more persistent branches start departing from the diagonal. This change signifies that clusters become apparent in the point cloud. The low persistent branches indicate very small point accumulations at different density levels. They occur in found clusters, and in regions where clusters are still combined. A further reduction of σ in Fig. 7d leads to more branches of high persistence and we observe that small branches accumulate in the lower part of the diagonal. This moment is a critical situation, as it tells us that separable regions indeed *exist*. If the point cloud actually consisted of one single cluster, the branches would just have moved from the top-right hand side to the lower bottom without departing from the diagonal. This behavior reflects that similarly distributed points cannot be separated into several dense regions, except with very small filter radii. Further reducing σ (Fig. 7e) leads to branches converging towards their final position on the ordinate. We note that they do not continue to move upwards, as the scale on the ordinate is decreasing all the time. A branch's final position depends on the multiplicity of the points' occurrences. In our system, we support multiple occurrences of points. That is, the input point cloud can have more than one point at the same position. As a result, these points have a minimum density depending on the number of duplicates. If there are, e.g., two points with the same coordinates, both have a density of, at least, two. The final diagram in Fig. 7f illustrates the other trivial case. The very small filter radius assigns each point its own density maximum, and the corresponding branches accumulate at their final position. The diagram indicates that there are many doublets and even some triplets in the example dataset.



(a) $\sigma = 30,000$

(b) $\sigma = 1,000$

(c) $\sigma = 100$

(d) $\sigma = 50$

(e) $\sigma = 10$

(f) $\sigma = 0.0001$

(g)

Fig. 7. (a)-(f) By continuously decreasing the filter radius, branches move along the diagonal and converge to their final position on the ordinate. If the data contains clusters, branches of high persistence depart from the diagonal. We search for a radius where many persistent branches depart from the diagonal. (g) Suitability diagram to summarize edge stabilities for different filter radii. We choose σ depending on where the plot features its local minimum.

Based on these observations, we can provide a guide-line for choosing an appropriate filter radius. We vary the value of σ between the two trivial cases while examining the persistence diagram. If branches depart significantly from the diagonal (relative to the noise) we have identified a good start radius which can then be improved by analyzing the hills' shapes in the profile. We support this visual exploration with our simplification controller (provided all thresholds are set to zero).

5.3 Filter Radius Suitability Diagram

Because persistence usually is a good indicator for cluster relevance, the visual determination of the filter radius via a persistence diagram already yields good results. However, the diagram ignores that spread clusters of lower persistence might also be relevant. To refine the criterion of a good start radius we aim to construct a function that measures the suitability of a particular filter radius. A plot of this function would then signify a suitable filter radius as the value of σ where the function changes. The function would also permit automatic filter radius determination by searching for the change algorithmically.

For the suitability, we directly use a cluster's stability. Note that stability is affected by persistence and cluster size, but also considers the point distribution. Therefore, stability is more precise than just the product of both values (equivalent to the relation of a hill's area to its circumscribing rectangle in the profile). To evaluate the suitability of a particular filter radius, we consider the corresponding density function's join tree, sum up all edge stabilities, and normalize this sum by σ. A plot is then obtained by calculating the suitability for different filter radii. Fig. 7g shows the plot for the image segmentation dataset. Due to σ-normalization, the plot features very large values for small radii, and small values for large radii. Our desired filter radius is characterized by the local minimum of the function, which is approximately at $\sigma = 35.0$ in this example.

We provide the user with an interactive suitability diagram that already shows the plot for the smallest and largest possible values of σ (which is a line perpendicular to the diagonal). For the latter, we can use a multiple of the dataset's diameter. Automatic determination of the function's change is achieved by evaluating different radii on the logarithmic scale either equidistantly or using a standard divide-and-conquer approach between the two initial values. In both cases, the plot is locally refined at the position where the first minimum is found, i.e., where an evaluation leads to a smaller value than at its two neighboring evaluations. The result might be a plot (for visual inspection) or the value where the refinement stopped. The user can also refine manually.

6 Evaluation & Demonstration

We compare our topological landscape profile to competing visualization techniques, and we discuss their power to characterize a clustering in terms of hierarchy, compactness and extent. Furthermore, we demonstrate how the global topological overview already indicates interesting features for subsequent inspection via traditional techniques like PCA and PCP.

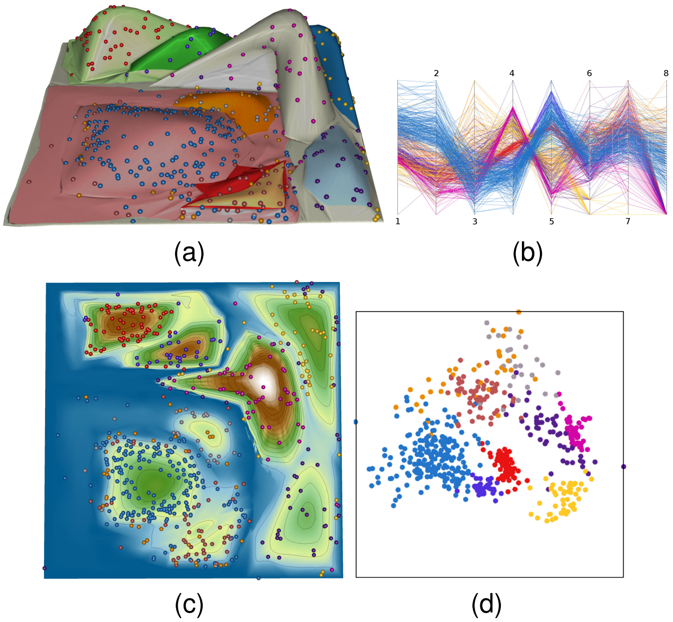
The time needed to present a final landscape profile depends on three steps: the topological analysis, identifying appropriate parameters, and constructing the profile. Note that finding an appropriate filter radius usually implies running the topological analysis several times. Because the actual runtime of the topological analysis depends on the chosen neighborhood graph and several optimization steps, we refer the interested reader to [6] for details. Most of the time is spend on constructing the approximation of the high dimensional density function. The construction of the join tree is very fast ([7]). Simplifying the join tree, extracting the branch decomposition, and constructing the profile can be considered as operators on the join tree. That is, they are generally fast, but depend on the tree's complexity. In practice, the analysis of some ten-thousand points in around fifty dimensions generally is a matter of seconds on our machine with two 2.6 GHz Quadcore processors and 8 GB memory. We implemented our prototype under Linux in C++, but do not yet utilize GPU acceleration (the join tree calculation on the GPU is an open research topic). The profile constructions (including the topological analysis) for the examples in this section took less than one second each.

6.1 Visualization aspects

We consider the Italian olive oils dataset [41]. It consists of 572 oils from nine different regions in Italy. The features describe the percentage of eight fatty acids. We try to analyze whether these oils form clusters based on their combination of fatty acids. Common clustering information usually includes the number of clusters, sub-cluster hierarchy, and cluster properties (such as size, or compactness). A useful cluster visualization should permit easy extraction and comparison of this information. Fig. 8 and Fig. 9c show several visualizations of this dataset (parameter choice is given by Figs. 9a,b). We do not consider simple hierarchy trees, as they only focus on this particular property of a clustering.

The landscape profile in Fig. 9c suggests three separated clusters, each with sub-clusters. We can easily identify and compare cluster compactness and size. For example, the "Umbria/Liguria" cluster is the most compact one, while the cluster on the right has significantly more points, but is less compact. Together with the classification information, conveyed by the histograms, we notice that clusters, and thus the combination of fatty acids, correspond to the major growing areas in Italy.

A partition of the results in geometry-based (Fig. 8b,d) and topology-based (Fig. 8a,c and Fig. 9c) visualizations shows that both PCP and PCA heavily focus on the data points and their color, while the topological visualizations focus on structure and only augment the data. For



(a) (b) (c) (d)

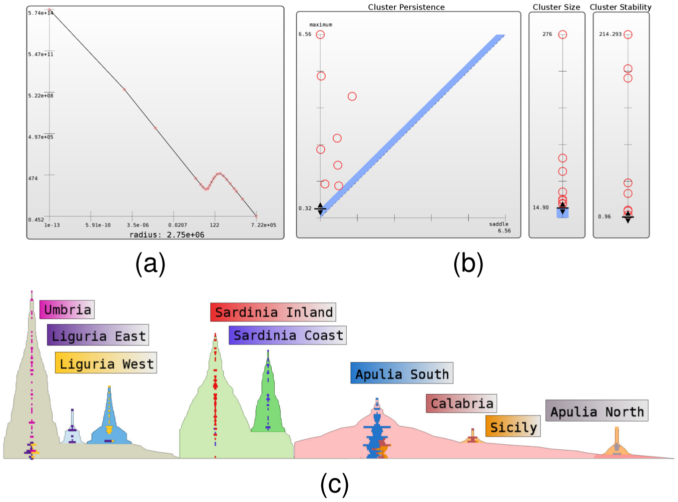
Fig. 8. Competing visualizations for the olive oils dataset: (a) 3D topological landscape, (b) parallel coordinates plot (PCP), (c) flattened 3D topological landscape, and (d) projection onto the first two principal components (PCA).

unclassified data, this implies that PCP and PCA can likely not answer basic clustering questions if points are just black. The topological visualizations, however, would still provide the same structural insights.

Considering the PCA in Fig. 8d, even for classified data, extracting the same clustering information can be infeasible. Because projections focus on properties like inter-cluster distance and shape, which are not necessarily needed to describe a clustering, they are less appropriate to illustrate structure. For example, the ”Sardinia Coast“ and ”Apulia South“ points seem to constitute a cluster in the projection. However, this is only an illusion caused by the 30% projection error. The profile tells us that these points correspond to two different clusters. Cluster compactness might also be an artifact and occlusion prevents the user from manually counting data points to determine and compare cluster sizes.

The same holds for overviews via a PCP (Fig. 8b). Even for colored data, neither basic clustering structure nor sub-clusters are easily perceivable. To extract and compare cluster properties, poly-lines need to be count and analyzed throughout all dimensions.

Coming back to topology-driven cluster visualizations, Fig. 8a shows the original 3D topological landscape [24]. Obviously, proper feature comparison is complicated by view-dependent occlusion, invisible data points, perspective distortion, and degenerated footprints. Furthermore, a hill's height and footprint are not simultaneously observable. Harvey *et al.* [25] improved the degenerated footprints using tree-maps. However, for very different ratios, this still does not facilitate exact visual comparison and also suffers from problems in 3D.



(a) (b) (c)

Fig. 9. Olive oils dataset: (a) the filter radius controller and (b) the simplification controller are used to determine the parameters necessary to obtain an appropriate (c) topological landscape profile

The atoll-like visualization in Fig. 8c (which is basically a colored version of Fig. 8a, seen from above) eliminates occlusion problems, but still makes feature comparison difficult, especially regarding the footprints.

The question arises whether there are tasks that favor one of the previous visualizations. Clearly, geometric approaches are only useful if the dataset is either simple, or if it can be simplified, e.g. by using hierarchical parallel coordinates [11]. Therefore, we also use them for local analysis instead of the global overview. We also note that the 3D landscapes as proposed by Weber *et al.* and Harvey *et al.* originate from other application domains with more topological complexity. That is, they are naturally 3D, but unnecessarily suffer from it when used for clustering purposes which do not require 3D. However, they still have advantages: First, a better screen utilization. When observed from above, the quadratic shape of the whole landscape is more compact than our left-to-right layout in 2D. This compactness in combination with the hills' 2D footprints also allows to visualize more features on the same screen compared to our profile.

6.2 Data Analysis aspects

In the style of *structure-based brushing*, as proposed by Fua *et al.* [5], we now demonstrate how the topological perspective, conveyed by the landscape profile, signifies features in the data that are worth further analysis. These features would likely be occluded without the structural perspective onto the data. We use the image segmentation dataset [36] again.

Fig. 10a and Fig. 10b illustrate the whole dataset via PCA and PCP, respectively. Instead of recalling their drawbacks for large, high-dimensional data, we rather want to utilize our landscape profile, shown in Fig. 10c, to reveal and further inspect features that are

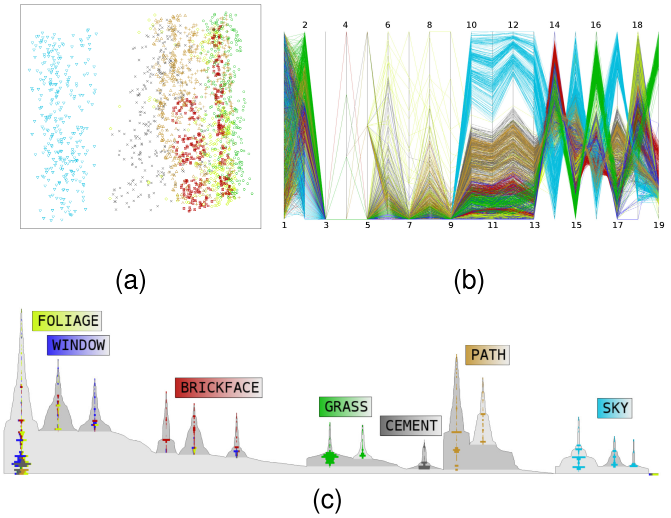


Fig. 10. Image segmentation dataset: comparison between (a) PCA projection, (b) parallel coordinates plot, and (c) the topological landscape profile. Data point colors reflect class affiliation. Due to its focus on structure, the profile permits convenient clustering identification and cluster comparison - even for unclassified data.

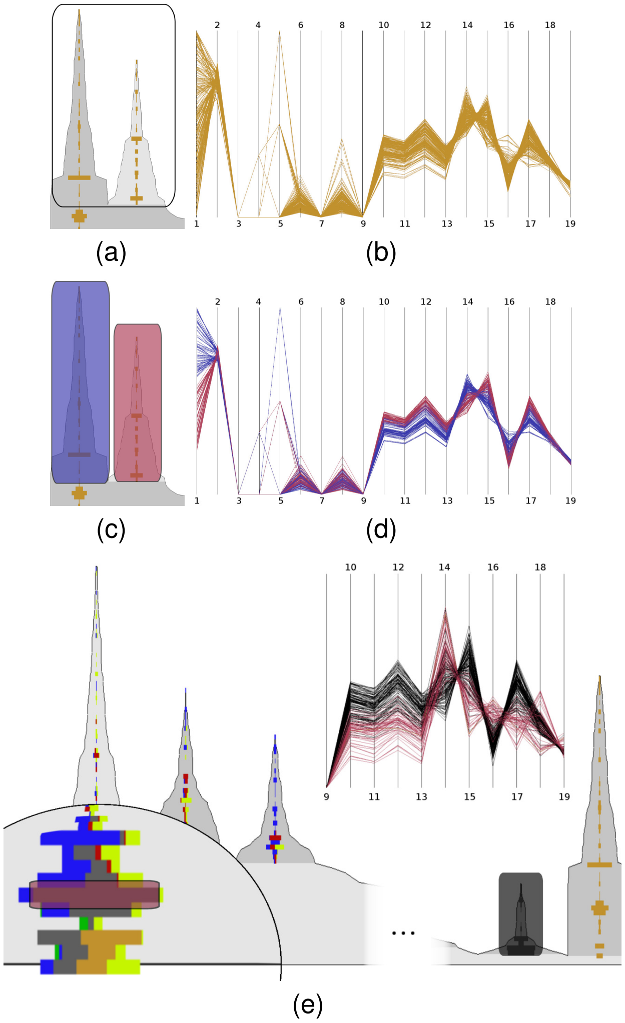


Fig. 11. (a)-(d) Brushing single hills and sub-hills in the profile not only reduces visual clutter in the PCP, it also explains *why* clusters have sub-clusters. (e) A class-wise selection of the histograms can be used to analyze why points of the same class are not in the same cluster.

not readily obvious in PCA and PCP. The landscape profile illustrates clusters of different relevance (in terms of compactness, coherence and the number of points), overall consistent with the data's classification, and it suggests that the SKY cluster can be separated from the rest. At least this separation is also suggested by Figs. 10a,b. We use our overlap-free visualization to focus on several data features. For example, we quickly realize that, e.g., the PATH hill is much higher than the SKY hill. A delegation of only these points to the PCP-view quickly confirms that the brown poly-line bundle is generally more compact, especially in the subspace spanned by the 10^{th}-13^{th} dimensions (Fig. 10b). As these dimensions encode RGB values in the original images, this diversity of SKY points suggests sub-clusters for different kinds of sky. To isolate a single cluster, we either select the histograms on the cluster's hill (Fig. 11a), or we click on the hill(s). As shown in Fig. 11b, the global trend of the selected points in the PCP is now much clearer and easier to observe. The sub-hills in the profile furthermore indicate that this already compact poly-line bundle is even more separable. To investigate the reasons for this, we link each hill individually with different colors (Fig. 11c). The PCP in Fig. 11d reveals why the cluster breaks into sub-clusters: although the points have the same global trend, in some dimensions they are widely spread (D1), shifted (D10-D13, D16-D17) or even inverted (D14-D15).

Another feature, that is not so obvious at first, is indicated by the histograms on the profile's leftmost hill (Fig. 11e). While gray CEMENT points have their own cluster in the middle of the profile, they additionally accumulate at lower height on the main hill. To analyze this phenomenon, we select the said histogram bars and the CEMENT hill individually with different colors. The PCP confirms that although being in the same class, the corresponding points differ pretty much between the 9^{th} and 17^{th} dimension and are thus in separated clusters.

In addition to this subspace analysis via axis-based techniques, a cluster's shape, extent or distance to other clusters may also provide relevant insights. Again, we use our global knowledge to further improve the projection result. For example, if we wanted to improve the projection of the cyan SKY points (that only account for around 84% of their original variance in Fig. 10a), we could easily delegate them to the PCA-view by clicking on each of the three hills (cf. Fig. 12a). This increases the explained variance to approximately 92% in Fig. 12b and also separates the sub-clusters visually in the projection - even if no classification was available. The histograms

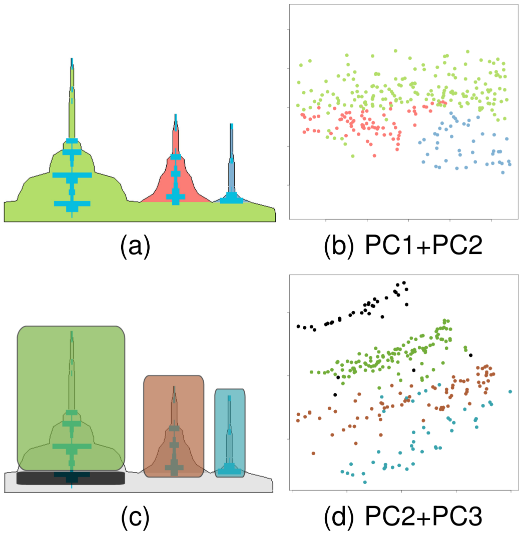


Fig. 12. (a)-(b) A hill-wise selection of (sub-)clusters increases the projection quality and separates the points visually (even for unclassified data). (c)-(d) If the profile indicates more features than the projection can reflect, the result can be improved by individual selections and adjusting the projection.

and the shape of the leftmost SKY hill suggest that even more groups could be separated. Although the first two principal components account for around 92% of the cyan points' original variance, the first three principal components account for almost 99,3% variance. Because the green points in Fig. 12b do not seem to reflect a further separation, we might suffer from projection artifacts and this separation must be hidden in the third principal component. In fact, if we select the histograms with different colors (Fig. 12c) and project the data onto the second and third principal component, we obtain a projection that reflects this separation. PCA did not choose this perspective because it features less data variance than Fig. 12b. However, the landscape profile already suggested further investigation if we were interested in maximizing separability in the final image.

7 Conclusion & Future Work

Because projections and axis-based techniques suffer from information loss, projection artifacts, occlusions, and visual complexity, we proposed the split the visual analysis of high-dimensional clusterings into two separated phases. In the global overview phase, we neglect geometric properties to allow an overlap-free presentation of the clustering in terms of the clusters' number, hierarchy, compactness, size, and variance. We use previously introduced topology-/ density-based analysis to identify the global clustering, but present this knowledge in a novel landscape profile from which structure can be extracted more precisely and with less user interaction efforts. To help users to choose parameters conveniently, we presented scented, interactive controllers.

In the local analysis phase, the topological perspective on the data is used to brush-and-link features to other views. Because only selected points need to be handled, common techniques can reduce information loss, illusions, and visual clutter. This reveals knowledge that was hidden before. The global knowledge also permits to classify unclassified data based on the association of points to their dense regions. Our future endeavors concern the capturing and representation of other topological or even geometrical cluster properties.

Acknowledgments

The authors would like to thank anonymous reviewers for valuable comments and assistance in revising the paper. The work presented in this paper was supported by a grant from the German Science Foundation (DFG), number SCHE663/4-1 within the strategic research initiative on Scalable Visual Analytics (SPP 1335), and by the the Department of Energy (DOE) Office of Science, Advanced Scientific Computing Research (ASCR), under Contract No. DE-AC02-05CH11231.

Disclaimer

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

References

[1] A. Inselberg and B. Dimsdale, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," in *VIS '90: Proceedings of the 1st conference on Visualization '90*, 1990, pp. 361–378.

[2] W. Cleveland and M. McGill, *Dynamic graphics for statistics*. Wadsworth & Brooks/Cole Advanced Books & Software, 1988.

[3] I. T. Jolliffe, *Principal component analysis*. Springer, 2002.

[4] J. C. Roberts, "State of the art: Coordinated multiple views in exploratory visualization," in *Proceedings of the 5th International Conference on Coordinated Multiple Views in Exploratory Visualization (CMV2007)*. IEEE Computer Society Press, July 2007.

[5] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner, "Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, pp. 150–159, 2000.

[6] P. Oesterling, C. Heine, H. Jänicke, G. Scheuermann, and G. Heyer, "Visualization of high-dimensional point clouds using their density distribution's topology," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 1547–1559, 2011.

[7] H. Carr, J. Snoeyink, and U. Axen, "Computing contour trees in all dimensions," *Computational Geometry*, vol. 24, no. 2, pp. 75–94, 2003.

[8] T. Kohonen, *Self-Organizing Maps.*, 3rd ed. Springer, 2001.

[9] J. B. Kruskal and M. Wish, *Multidimensional Scaling*. SAGE Publications, 1978.

[10] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings of the 1996 IEEE Symposium on Visual Languages*. IEEE Computer Society, 1996, pp. 336–343.

[11] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner, "Hierarchical parallel coordinates for exploration of large datasets," in *Proceedings of the conference on Visualization '99: celebrating ten years*. IEEE Computer Society Press, 1999, pp. 43–50.

[12] J. Yang, M. O. Ward, and E. A. Rundensteiner, "Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets," *Computers & Graphics*, vol. 27, no. 2, pp. 265–283, Apr. 2003.

[13] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey, Eds., *Graphical methods for data analysis*. The Wadsworth Statistics/Probability Series, 1983.

[14] R. A. Becker and W. S. Cleveland, "Brushing scatterplots," *Technometrics*, vol. 29, no. 2, pp. 127–142, 1987.

[15] J. Johansson, P. Ljung, M. Jern, and M. Cooper, "Revealing structure within clustered parallel coordinates displays," in *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*. IEEE Computer Society, 2005, pp. 125–132.

[16] M. Novotny and H. Hauser, "Outlier-preserving focus+context visualization in parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 893–900, 2006.

[17] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner, "Visual analysis of large graphs: State-of-the-art and future research challenges," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1719–1749, 2011.

[18] J. Han and M. Kamber, *Data mining: concepts and techniques*, ser. The Morgan Kaufmann series in data management systems. Elsevier, 2006.

[19] J. B. Kruskal and J. M. Landwehr, "Icicle plots: Better displays for hierarchical clustering," *The American Statistician*, vol. 37, no. 2, pp. 162–168, 1983.

[20] T. Lindeberg, "Scale-space theory in computer vision," 1994.

[21] A. Hinneburg and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *Knowledge Discovery and Data Mining*, 1998, pp. 58–65.

[22] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," in *Proceedings ACM SIGMOD International Conference on Management of Data*. ACM Press, 1999, pp. 49–60.

[23] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, 1996, pp. 226–231.

[24] G. Weber, P.-T. Bremer, and V. Pascucci, "Topological landscapes: A terrain metaphor for scientific data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1416–1423, 2007.

[25] W. Harvey and Y. Wang, "Topological landscape ensembles for visualization of scalar-valued functions," *Computer Graphics Forum*, vol. 29, no. 3, pp. 993–1002, 2010.

[26] B. Shneiderman, "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Trans. Graph.*, vol. 11, pp. 92–99, 1992.

[27] P. Oesterling, G. Scheuermann, S. Teresniak, G. Heyer, S. Koch, T. Ertl, and G. H. Weber, "Two-stage framework for a topology-based projection and visualization of classified document collections," in *IEEE Conference on Visual Analytics in Science and Technology (IEEE VAST)*. IEEE Computer Society, 2010, pp. 91–98.

[28] S. Takahashi, I. Fujishiro, and M. Okada, "Applying manifold learning to plotting approximate contour trees," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1185–1192, 2009.

[29] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, December 2000.

[30] S. Gerber, P.-T. Bremer, V. Pascucci, and R. Whitaker, "Visual exploration of high dimensional scalar functions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 1271–1280, 2010.

[31] W. Willett, J. Heer, and M. Agrawala, "Scented widgets: Improving navigation cues with embedded visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1129–1136, 2007.

[32] C. Ahlberg, C. Williamson, and B. Shneiderman, "Dynamic queries for information exploration: An implementation and evaluation," in *CHI*, 1992, pp. 619–626.

[33] G. Jaromczyk, J.W.; Toussaint, "Relative neighborhood graphs and their relatives," in *Proc. of the IEEE*, ser. Issue 9, vol. 80, 1992, pp. 1502–1517.

[34] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," *Discrete & Computational Geometry*, vol. 28, no. 4, pp. 511–533, 2002.

[35] N. Kong, J. Heer, and M. Agrawala, "Perceptual guidelines for creating rectangular treemaps," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 990–998, 2010.

[36] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[37] J.-D. Fekete and C. Plaisant, "Excentric labeling: dynamic neighborhood labeling for data visualization," in *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, 1999.

[38] V. Pascucci, K. Cole-McLaughlin, and G. Scorzelli, "The toporrery: computation and presentation of multi-resolution topology," in *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, ser. Mathematics and Visualization. Springer, 2009, pp. 19–40.

[39] H. Carr, J. Snoeyink, and M. van de Panne, "Simplifying flexible isosurfaces using local geometric measures," in *Proceedings of the conference on Visualization '04*, ser. VIS '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 497–504.

[40] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," *Discrete Comput. Geom.*, vol. 37, pp. 103–120, 2007.

[41] M. Forina, C. Armanino, S. Lanteri, and E. Tiscornia, "Classification of olive oils from their fatty acid composition," *Food Research and Data Analysis*, pp. 189–214, 1983.