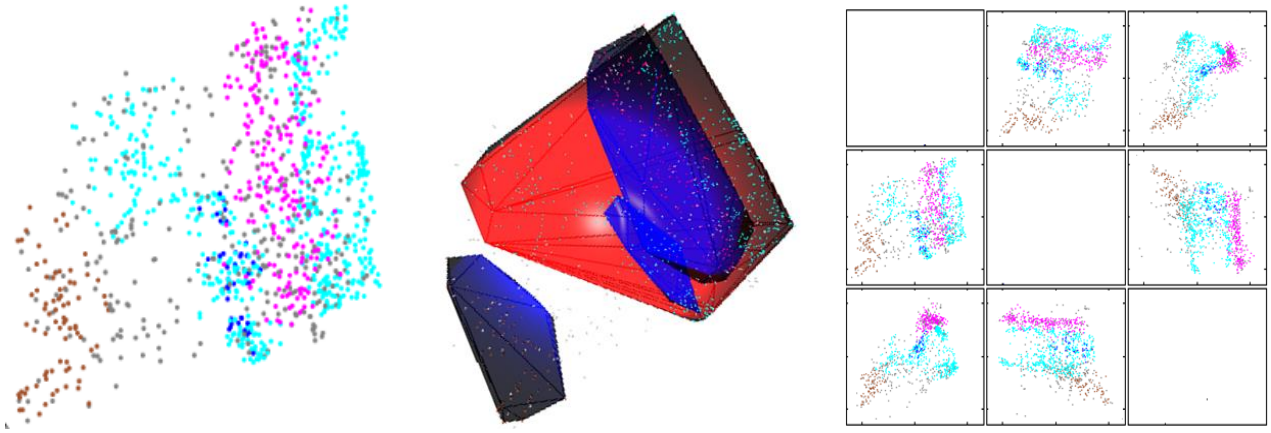
# Does 3D Really Make Sense for Visual Cluster Analysis? Yes!

Bing Wang, Klaus Mueller

Visual Analytics and Imaging Lab, Computer Science Department, Stony Brook University



(a) 2D scatterplot (b) Shaded 3D display (c) SPLOM

Figure 1. 2D scatterplot vs 3D display vs SPLOM.

**ABSTRACT**

Our paper takes the stance that a 3D shaded display can add a significant amount of information to the visualization of high-dimensional data. We believe that it makes better use of the innate cognitive capabilities of the human visual system which is highly optimized to recognize and reason with 3D shape information. As a first step we studied a variety of real-world datasets and confirmed that the extension from the traditional 2D space to 3D space is indeed justified – most datasets we studied had clusters residing in subspaces with more than two significant principal components. We then describe an interactive interface that allows users to navigate these 3D subspaces, expand the exploration to higher dimensionalities, and also transition among the distinct subspaces inhabited by different clusters in the data.

**Keywords:** Multivariate data, high dimensional data, subspace clustering

## 1 INTRODUCTION

Does 3D really make sense for data visualization? Or, more specifically, does it make sense for visual cluster analysis, which is a sub-field of visualization? This principal question might be framed in visual perception theory. It was Hermann von Helmholtz, who in the 19th century performed the first modern study of visual perception. When von Helmholtz examined the human eye he concluded that they could not aid humans in the perception of 3D shapes directly. And indeed, as was already suspected by Helmholtz, it was later scientifically shown [6] that human perception of the 3D physical world we live in is learned during infancy. During this time an unconscious inferential chain is established which is used to transform the input coming from the eye's optical system into the perception of 3D shape and relations. These neural circuits can not only make inferences about 3D shapes and topologies, they can also resolve complex patterns and textures. So one may ask, why not take advantage of this complex neural circuitry, either by ways of stereo vision or motion parallax. Especially the latter is an interesting concept since we can easily facilitate it on the computer, via interaction, without the need for special glasses. It is also how we perceive 3D objects further away. And finally, we can also use other depth and shape cues, such as shading, shadows, depth of field, transparency, and the like, and control them via interaction.

We have begun developing an interactive framework and system that capitalizes on these concepts. The first instantiation of our research gave rise to the TripAdvisor$^{ND}$ system [10]. It provides a touchpad-like interface by which users can smoothly tilt the projection plane in high-dimensional space to produce multivariate scatterplots that best convey the data relationships under investigation. These *dynamic scatterplots* appeal to the human cognition of 3D textures via motion parallax.

In our current work we have continued on this path, but now addressing the notion of high-dimensional shape. In this context, *shape* is the high-dimensional manifold covering a point distribution (or cluster). Our system breaks this high-dimensional manifold into 3D sub-spaces which innately appeal to the human cognition of 3D shape. Similar to TripAdvisor$^{ND}$ we also provide an interactive interface that allows users to change their viewpoints. But now we break the navigation into two modes. In the first mode users can only explore the current 3D space, while in the second they can transition to adjacent 3D spaces spanned by the original high-dimensional space.

* {wang12, mueller}@cs.stonybrook.edu

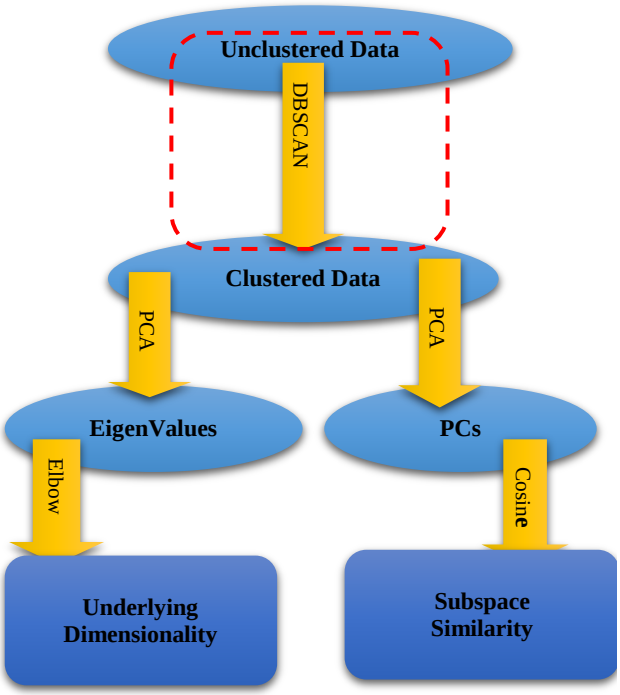



Figure. 2. Study workflow

To appreciate the power of 3D, let's consider Fig. 1 which compares a multivariate scatterplot (Fig. 1a) with a shaded 3D display (Fig. 1b), both taken from the same orientation. Points colored in grey are outliers. We find that the third dimension, in combination with the shaded display, allows users *to peek around* cluster shapes, mitigating the point overlaps that exist in the scatterplot projection. This already works well in a static display (given a good viewpoint) and it so alleviates the need for interaction to invoke motion parallax. Finally, Fig. 1c shows a 3×3 scatterplot matrix (SPLOM [5]) that allows users to look at all pairwise combinations of dimensions. While we have not formally tested this yet, we believe that our navigable 3D shape display might help users gain a better understanding of the shapes of clusters and their relationships, compared to the generalized scatterplot or the 3×3 SPLOM.

Our current system is primarily designed for cluster analysis, i.e., we do not assume any prior classification of the data. We consider each cluster a sub-space of the data. Here, a cluster is a set of ε-connected points where ε is the minimal distance a point must have to some other point in the cluster to also be part of this cluster. This property is not fulfilled by the k-means algorithm but is common in sub-space clustering. The outcome of this clustering is a set of sub-spaces and associated shapes, each of which has a certain intrinsic dimensionality (ID). The ID determines the complexity of the shape display needed to visualize it.

A classic method to discover the ID is Principal Component Analysis (PCA). If ID=2 then a conventional scatterplot will do. On the other hand, if ID=3 then a simple 3D display is sufficient. And finally, if ID>3 we need to allow users to transition between multiple 3D shapes, one for each distinct PCA vector set of three.

In this current research, we were particularly interested in finding out how appropriate the simple 3D display would be in practice. For this purpose we studied a variety of representative datasets to determine the ID characteristics of the subspaces in each. Subspace clustering can significantly reduce the ID of the data. It essentially decomposes the high-dimensional data into a composite of lower-dimensional independent phenomena, for which a 3D display or a transitional display that is not overly more complex might be sufficient. To find these subspaces we used the well-established DBSCAN clustering algorithm [4], augmented by a visual interface that gave us some insight into proper parameter settings to reach the different clusterings quickly.

The outline of our paper is as follows. In Section 2 we describe related work. Section 3 reports on the study we conducted to gain insight into the ID characteristics of a selection of datasets. Section 4 describes our sub-space visualization framework which also features our 3D display. Section 5 ends with conclusions and pointers to future work.

## 2 Related work

Subspace analysis is a rich topic and the various approaches are well summarized in the survey paper by Kriegel et al. [9]. While DBSCAN is not a sub-space clustering algorithm per-se, it can be used in conjunction with one, such as SUBCLU [8]. We chose DBSCAN because it gives clear definitions on how to join points into clusters of arbitrary shape. Subspace clustering can give rise to large collections of subspaces and methods for their efficient visualization and management have been proposed by various authors, such as Tatu et al.[14] and Yuan et al. [16]. Our research does not aim into this direction. Rather, users can steer a specific clustering and then represent each cluster as a separate sub-space with specific intrinsic dimensions.

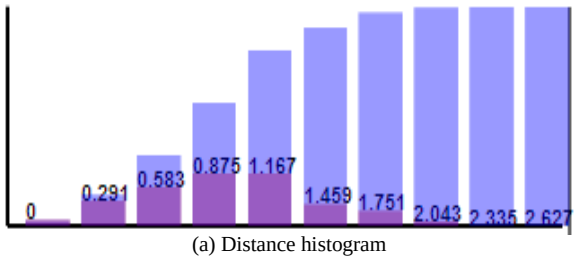
Sedlmair et al. [11] investigate the relationship of dimension reduction and visualization paradigm (2D and 3D scatterplots and SPLOMs) with regards to the ability of users to discern cluster separability. They find that 3D scatterplots do not provide additional benefits for the particular task they studied, but they also argue that 3D displays might be a good choice if the intent was to recognize cluster shapes. Our work has this intent, and in addition, our 3D display does not only allow users to interact with 3D scatterplots but also with 3D shapes represented as geometry.

Other methods that have exploited dynamic transitioning of scatterplots include ScatterDice [3] which restricts the transitions to motions between two SPLOM tiles, giving rise to a dynamic 3D point cloud projection display. Similar to our tool, the popular GGobi system [12], derived from the seminal concept of the 'Grand Tour' [1], also employs trackball controls but it does not have the advanced subspace exploration facilities our trackball interface provides. For example, with GGobi users cannot explicitly travel between adjacent subspaces and navigate the space via a dedicated map.

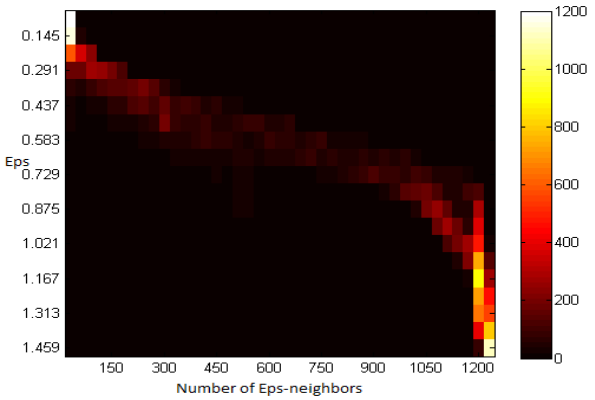
The iPCA framework of Jeong et al. [7] shows users how the original data dimensions contribute to both PCA space and the clustering. Users can interactively manipulate the contribution of each individual dimension and then observe the impact as transient changes in the scatterplot visualizations. We also make use of PCA, but we only do so for dimension reduction.

## 3 Dataset Study on Intrinsic Dimensionality

To determine whether a 3D display would suffice for visual cluster analysis, we conducted a series of studies on a variety of unclustered datasets, ignoring any classification when available. The workflow of our analysis is depicted in Fig. 2. As a first step, for each dataset, we performed density-based spatial clustering via DBSCAN to obtain a set of clusters of arbitrary shape. Here, the tuning of the DBSCAN parameters can give rise to different numbers of clusters (see Section 3.1). Next, we ran Principal Component Analysis (PCA) on each cluster and used the elbow method/scree plot to estimate their intrinsic dimensionality. We also compared the PCs of different clusters via the cosine similarity to determine if the clusters exist in the same or different subspaces. In the following sections we describe our study methodology in detail and discuss its results.




(a) Distance histogram




(b). Heat map describing the number of points that have a certain number of ε-neighbors

Figure. 3. DBSCAN visualizations

### 3.1 DBSCAN

DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise and is one of the most widely used and cited data clustering algorithms. Its key concept is to define clusters based on the notion of *reachability*. Gven two points, $p$ and $q$, if the distance between them is less than ε and $q$ has a sufficient number of neighbors within the ε distance, we say $p$ is *directly density-reachable* from $q$. On the other hand, $p$ and $q$ are *density-reachable* if there exists a sequence of points $p_1, p_2, p_3 \ldots p_n$ where $p_{k+1}$ is directly density-reachable from $p_k$ ($k = 1,2,3 \ldots n$-1), then $p_1 = p$ and $p_n = q$. Finally, if there is a third point $r$ from which both $p$ and $q$ are density-reachable, $p$ and $q$ would be *density-connected*. Every point-pair inside one cluster found by DBSCAN must be density-connected, and if a point is density-reachable from any point within one cluster, it also belongs to that cluster.

DBSCAN requires two parameters: the neighborhood radius ε and the minimum number of points (*minPtn*) that a cluster should at least have. It also uses a flag to distinguish whether a point has already been processed or not. DBSCAN starts with an arbitrary yet unvisited point and finds all points that are no further than ε to it. If this number of points is greater than *minPtn*, a new cluster is started else the point is classified as noise. If a cluster is formed, all discovered points are added to the starting point's neighbor list. Next, for every point in the list (note that the elements of the list are dynamically added), its ε-neighborhood is also retrieved. If it is also dense (the number of points being larger than *minPtn*), all of its ε neighbors are also added to the list. This process continues until no density-connected points can be further discovered. Then DBSCAN finds the next unvisited point and repeats this process.

DBSCAN has many advantages over k-means clustering that is commonly used in visual analytics. The main advantage is that it retrieves high-dimensional structures defined by density and connectivity and not by radial distance to a centroid. As a consequence, it can find irregularly shaped clusters that are more descriptive of the underlying phenomena. Further, it is also robust to noise, and it does not require the number of clusters as input.

#### 3.1.1 Visual interface for DBSCAN parameter settings

DBSCAN is not parameter free – it requires users to choose the proper combination of ε and *minPtn*. DBSCAN is quite sensitive to these two parameters, but there is no general guideline on how to set them. And so, finding the settings that resulted in a defined change typically required much time consuming trial and error.

A first solution could be to run all possible setting as a background process and survey the clusterings that result. But this can take a considerable amount of time for reasonably sized datasets. Instead, we designed two visualizations that convey some idea about the relationships in the data and so provide some assistance in choosing the parameters.

The first of these visualizations is a distance histogram (Fig. 3a) which shows all pairwise distances between points. The purple bars are the normal histogram while the blue bars are the cumulative histogram which shows the setting at which the sharpest changes occur. These histograms convey the distance distribution of the data and allow users to pick specific ε-values that will likely give rise to a change in the clustering.

The second visualization is a 2D heat map (Fig. 3b) that visualizes the pairwise distances over the number of neighbors that are within each such distance. We constructed this plot by visiting each point, counting the number of neighbors for each discretized ε-setting, and incrementing the corresponding 'number of neighbors' bin of the plot. The plot allows users to estimate how many points would have a certain number of neighbors residing within a certain ε-distance, which can be helpful when choosing *minPtn* (and ε). In this particular example, we learn that the relationship is a fairly narrow curve, and so this plot saves users the considerable amount of time trying out *minPtn*-ε combinations that fall into the vast black areas of the plot.

### 3.2 Intrinsic Dimensionality Analysis

Following DBSCAN, we perform PCA on each discovered cluster. PCA uses an orthogonal transformation to find linear uncorrelated variables (the PCs) that describe the data. The strength of each PC vector – the eigenvalue – determines the amount of variation in the data it can explain. Normalizing these eigenvalues by the overall sum of eigenvalues expresses this strength in percent.

After obtaining these normalized eigenvalues and discarding those with values less than 0.001 we create a *scree plot* – an ordering of the eigenvalues from largest to smallest. The intrinsic dimensionality can then be estimated by locating the scree plot's *elbow* or *knee* – the point on the scree plot curve at which it stops to decrease significantly [13]. A simple metric to find this elbow is to draw a line from the first to the last point of the curve and then find the point that is farthest away from that line (see red circle in Fig. 4).

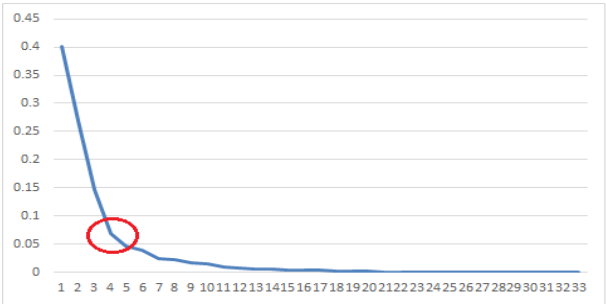
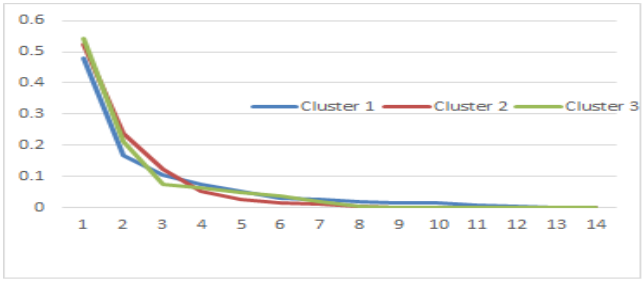
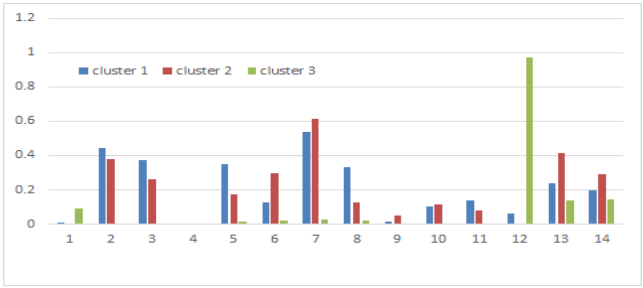



Figure. 4. Elbow method




(a) Scree plot




(b) Value histogram

| | Similarity (cluster 1 & 2) | Similarity (cluster 1 & 3) | Similarity (cluster 2 & 3) |
|---|---|---|---|
| PC1 | 0.91 | 0.15 | 0.13 |
| PC2 | 0.75 | 0.48 | 0.67 |
| PC3 | 0.56 | 0.38 | 0.49 |
| PC4 | 0.82 | 0.52 | 0.52 |

(c) Table: similarity between PCs

Figure 5. Boston housing dataset

While the elbow criterion provides a clear and deterministic way to decide the intrinsic dimensionality, we (and others [15]) found that often the elbow is not overly well expressed. The curve only slowly bends and a slight variation of the elbow metric can change its location drastically. Instead, it might be more appropriate to also look at the percent contribution of the eigenvalues. While we have not formally tested this, a contribution below a significance value of 0.05 (5%) may not account for much variation in the visual projection display. For the cluster plotted in Fig. 4 this would then point to an intrinsic dimensionality of 4-5.

Finally, we also conduct a similarity analysis of the subspaces found for a given dataset. We compute the cosine similarity for each significant PC pair for the two associated subspaces, and their normalized sum indicates if the two clusters reside in the same or similar subspace, or far apart. This then has implications on our subspace transitioning interface.

### 3.3 Results

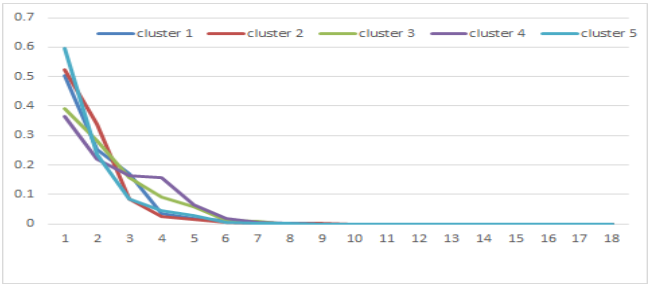
We studied a variety of datasets, most from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/). Below we present three representative results from this study. For each dataset, we first decide the intrinsic dimensionality of the clusters using the scree plot and then compute their cosine similarity.

#### 3.3.1 Boston Housing Data

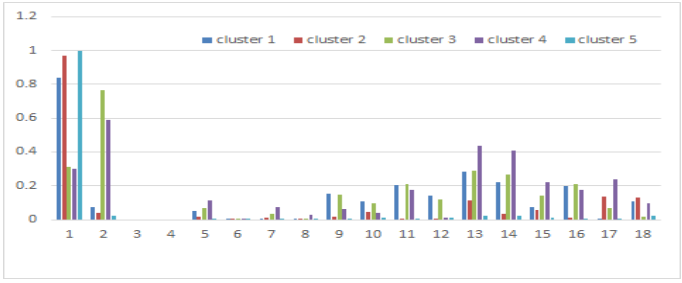
This dataset [18] describes housing values in suburbs of Boston. It has 506 instances with 14 continuous attributes. We set ε equal 0.5828 and *minPtn* to be 12. After running DBSCAN, we obtained three clusters. Fig. 5a shows the corresponding scree plot.

We observe that for all three clusters the amount of variance covered by the third PC dimension is at or above 5%, and it is at or above 10% for clusters 1 and 2. Only cluster 3 has a clear elbow at PC=3. Cluster 1 has it there as well, while cluster 2 has it at PC=4. Hence, a 3D display will be appropriate for all subspaces.

Fig. 5b shows the value histogram for PC1 and the table in Fig. 5c presents the cosine similarities between pairwise PCs. We observe that for clusters 1 and 2 the similarity of their most significant PC, PC1, is 0.91, while for the remaining three PCs, the similarity drops only slightly to 0.75, 0.56 and 0.82, respectively. On the other hand, PC1 for cluster 3 is quite different from the PC1 of




(a) Scree plot

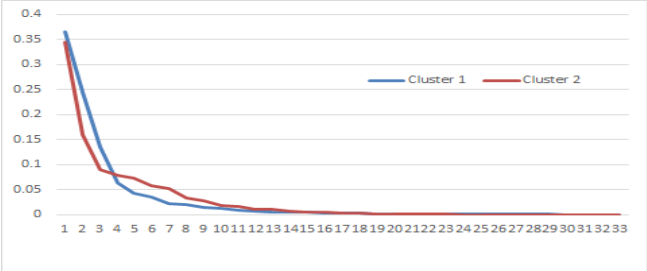



(b) Value histogram

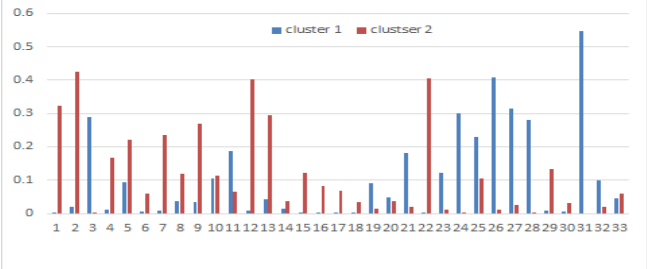
| Similarity / PC | Cluster 1 & 2 | Cluster 1 & 3 | Cluster 1 & 4 | Cluster 1 & 5 | Cluster 2 & 3 | Cluster 2 & 4 | Cluster 2 & 5 | Cluster 3 & 4 | Cluster 3 & 5 | Cluster 4 & 5 |
|---|---|---|---|---|---|---|---|---|---|---|
| PC1 | 0.89 | 0.60 | 0.62 | 0.86 | 0.40 | 0.44 | 0.97 | 0.92 | 0.34 | 0.24 |
| PC2 | 0.33 | 0.37 | 0.69 | 0.70 | 0.33 | 0.36 | 0.59 | 0.45 | 0.29 | 0.90 |
| PC2 | 0.79 | 0.48 | 0.82 | 0.34 | 0.67 | 0.57 | 0.57 | 0.52 | 0.62 | 0.31 |
| PC4 | 0.32 | 0.63 | 0.65 | 0.56 | 0.56 | 0.74 | 0.55 | 0.74 | 0.56 | 0.52 |

(c). Table: Similarity between PCs

Figure. 6. Image segmentation dataset




(a) Scree plot




(b) Value histogram

| | Similarity(cluster 1 & 2) |
|---|---|
| PC1 | 0.143 |
| PC2 | 0.105 |
| PC3 | 0.276 |
| PC4 | 0.305 |

(c) Table: Similarity between PCs.

Figure 7. ISDAC dataset

the other two clusters (0.15 and 0.13, respectively), and the remaining PCs also only have a similarity of about 0.5 with those of cluster 1 and 2. We hence conclude that (1) cluster 3 resides in a rather different subspace than cluster 1 and 2, and (2) the subspaces of cluster 1 and 2 are somewhat closer.

### 3.3.2 Image Segmentation Data

This dataset [19] is composed of feature vectors derived from 1,200 3×3 image patches – 300 random instances each from four image classes (Brickface, Cement, Foliage, and Grass). The feature vectors have 19 attributes (dimensions) which are statistical measures of the images, such as region centroid, region pixel count, density, hue, and others. The third attribute 'region-pixel-count' is 9 for all instances. We removed it and are left with 18 all-numerical attributes. We set ε=0.321 and *minPtn*=47 and obtained five clusters.

From the scree plot shown in Fig. 6a we find that for all clusters the amount of variance covered by the third PC vector is in the range of 10-15%, the elbow is at the 4th eigenvalue and the 5% significance is reached at the 4th and 5th. So again, a 3D display with transitioning capabilities will be helpful. – a single 2D projection will not be able to capture the variances sufficiently.

Fig. 6b shows the value histogram for PC1 and Fig. 6c presents the cosine similarities between pairwise PCs. Here we observe that probably the most similar clusters are cluster 1 and 5 as they have the most consistent PC vector similarities. Other clusters seem quite disparate.

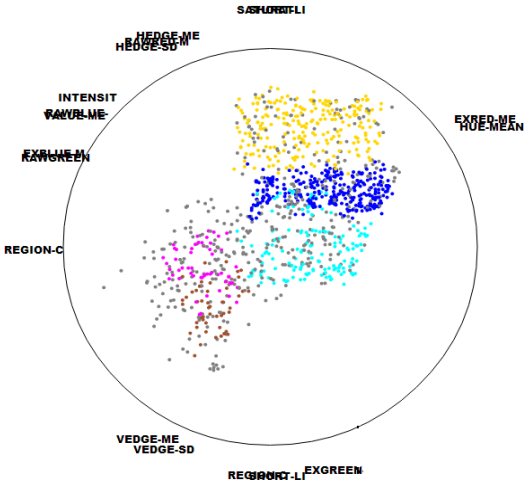



Figure. 8. Local subspace explorer.

### 3.3.3 ISDAC data

The ISDAC dataset [17] is an atmospheric dataset fused from multiple sources and consists of 221 data points, each a 33-dimensional vector composed of latitude, longitude, altitude, time stamp, temperature, and pressure and measurements on the cloud particles (cloud droplets presence, cloud particle concentration, etc.) and on aerosol particles (size and composition: soot, sulfate levels, organics, dust, sea salt, etc.). We set ε=0.7592 and *minPtn*=6 and obtained two clusters. Fig. 7 shows the results we obtained. We notice that at least three PCs are required to capture 90% of the data variance. The PCs for the two clusters are quite different (low cosine similarity values and very different PC1 value histograms) and hence the two clusters belong to entirely different subspaces.

## 4 Subspace Visualizer

The subspace visualizer is an interactive system to support the exploration of the subspaces found in the analysis phase. It is able to visualize the data points both as dynamic scatterplots and tessellated into solid shapes.

After running DBSCAN and obtaining the cluster information, all data points are shown in our *local subspace explorer* (LSE, Fig. 8). By default, the LSE displays the data in the coordinate space spanned by the three most significant PCs of the first cluster DBSCAN finds. The LSE has an integrated trackball interface that lets users transform the 3D subspace and look for interesting patterns. Additional trackball interactions are provided that allow users to move to adjacent 3D subspaces according to the cluster's PC configuration, if its intrinsic dimensionality is greater than 3.

Our system also provides another panel – the *global subspace explorer* (GSE, Fig. 9). This display has one central view surrounded by several smaller peripheral views. With the GSE the user can gradually transition from one subspace to another simply by dragging the mouse towards the desired peripheral view. The user can then inspect it more closely in the central display via the trackball functionality. For the current application, the peripheral views hold the subspaces of the other clusters found by DBSCAN.

Both the LSE and the GSE display allow users to highlight the cluster that matches the subspace currently selected for trackball-based exploration. Other clusters can then be either all colored in grey, another color, or in their assigned colors.

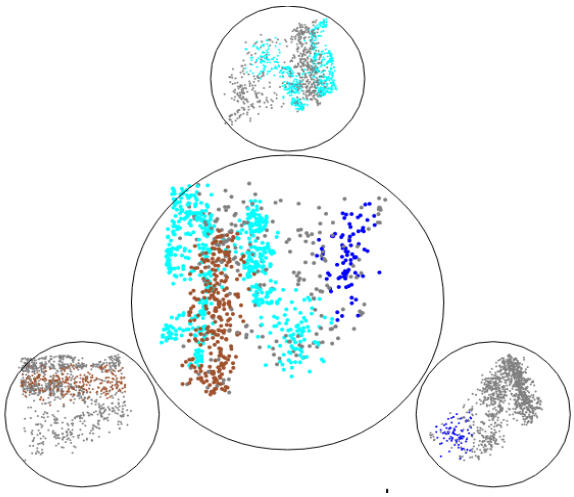
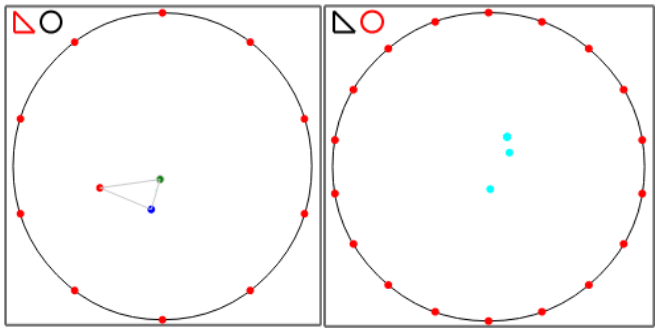


Figure. 9. Global subspace explorer.



(a) Triangle representation (b) Point representation

Figure 10. Subspace trail map.

A final component of our system is the *subspace trail map* (STM, Fig. 10) which shows the explored subspaces as points or triangles in context of the dimensions. The STM is useful to visualize the (dis)similarity of the various subspaces, to log and store newly found views and subspaces, and also to navigate them.

### 4.1 Local Subspace Explorer (LSE)

The key idea behind the local subspace explorer is the use of a 3D mouse trackball system (Fig. 11). Upon a mouse click, the screen location of the click is mapped onto a virtual unit sphere representing the trackball. This virtual sphere encapsulates the current generalized 3D subspace. We call it *generalized* because the axes are not necessarily dimension vectors. From the position of the last mouse click and the present one we can construct a plane with normal vector *n* and also compute the rotation angle$\theta$. From these two quantities the 3×3 rotation (or transformation) matrix *T* is derived (for more detail see [2]). In our case, we deal with high-dimensional point clouds and not with 3D objects, and so we require their projection into 3D trackball space before rotating with *T*. We achieve this by post-multiplying the trackball rotation matrix with the 3×*N* projection matrix *P*. The first two of these vectors (we call them *projection plane axis* (PPA) x-axis and y-axis) are the most two significant PCs we obtained when performing PCA for the first cluster. The third vector is the third most significant PC vector. This represents the PPA z-axis. Multiplying all these matrices generates the compound matrix *M*.

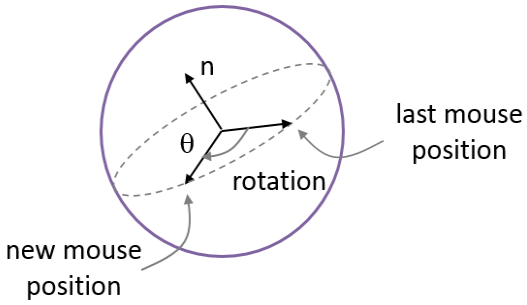



Figure. 11. 3D trackball concept.

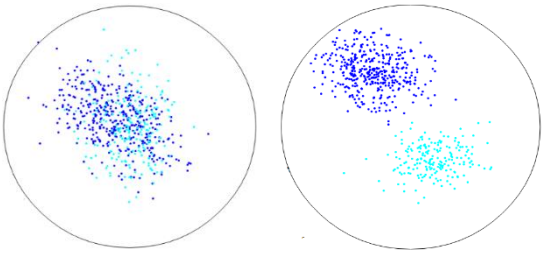


Figure. 12. Chase clusters.

There are three interactions one can perform with the LSE. We provide three types of operations all controlled with different mouse buttons depressed, described as follows.

#### 4.1.1 Explore a generalized 3D subspace.

This is the basic interaction performed by moving the mouse and the left mouse button depressed. The user 'grabs' on to the trackball and rotates the generalized 3D subspace via the compound projection matrix *M* as described above.

#### 4.1.2 Chase clusters in adjacent 3D subspaces.

One of the advantages of the dynamic display is that it allows one to quickly change the influence that a dimension has on the projection of the point cloud. Tilting the projection plane more into the dimension axis will spread the points along this dimension and so expose possible gaps between clusters or cluster components. An example to illustrate this concept is shown in Fig. 12. In the basic trackball interaction users may discover such an opportunity but it might be out of reach with the current generalized 3D subspace. In this case, with our interface, the user would let go of the left mouse button and instead press the right mouse button and move the mouse in the direction of this dimension's projection, as indicated by a text string at the trackball's periphery. The further the mouse is moved the more the projection plane is tilted into the dimension's axis vector. Conversely, moving backwards along that direction, towards the center of the trackball, will decrease the influence of this dimension.

#### 4.1.3 Go deeper into higher-dimensional space

A click on the middle mouse button will update the PPA z-axis to one of the less significant PCs, in order. Or users can also choose to let the system randomly generate one PPA z-axis. For the latter situation, based on the current PPA-x and PPA-y vectors – the first two rows in the compound matrix M – a new orthogonal vector is computed using Gram-Schmidt orthogonalization. This operation essentially moves the local 3D generalized subspace deeper into the high-dimensional universe.

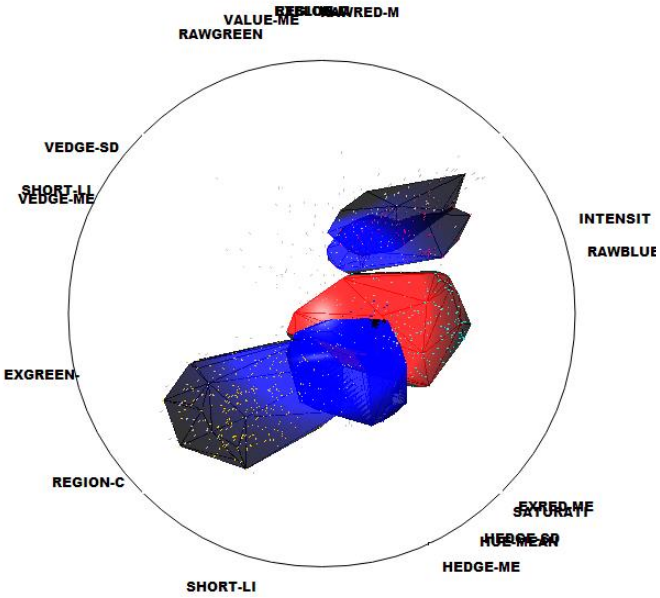



Figure. 13. Shape visualizer.

#### 4.1.4 3D Shape visualizer

The shape visualizer first tessellates each cluster's point cloud into a polygonal hull [2] and then displays it as a 3D geometric solid (Fig. 13). A light source is enabled and each solid is lit and shaded such that its shape can be visually well appreciated. The same coloring rules as for the points display are enabled also for the shape display. For example, in Fig. 13 the cluster native to the current subspace is highlighted in red. Highlighting the shape reminds users that this cluster is the one whose extent is most accurately represented by this subspace – all others have distribution extents that are possibly not or only partially covered in the current PCA axis configuration.

The surfaces of the solids are rendered semi-transparently such that the user is still able to see the data points in their interior. Finally, when the trackball is moved, the 3D shapes also move, allowing the user to gain a deeper understanding about the shapes of the clusters and their spatial relationships.

### 4.2 Global Subspace Explorer (GSE)

The core part of the GSE is the lens-shaped display in the center. It is surrounded by smaller peripheral lens-shaped displays which hold what we call the *key views*. Double-clicking on the center display pops-up the LSE which is the trackball. The number of small views is the same as the number of clusters. Each small view represents one cluster and its 3D vectors are this cluster's three major PCs. One can change the central view by either drag and drop one of the peripheral views or by view interpolation. Interpolation can be useful to discover interesting subspaces between these views.

### 4.3 Subspace Trail Map

To keep track of the history of subspaces explored so far we provide what we call the 'subspace trail map'. It maps the three subspace axis vectors into a circle in which the nodes representing the N dimensions are equally spaced on its perimeter. For each subspace, the vectors can either be drawn as a triangle (see Fig. 10a) or they can be averaged into a point (as seen in Fig. 10b) by switching between the top left '◯' or '△' icons. The latter display leads to less clutter and gives a better overview. The trail map is updated as the subspaces are changed so users can gain a bird's eye view on their voyage through high-dimensional space.

Every time when we obtain the subspaces for all clusters, we take the first three major PCs, do the above computation and map the subspace into the trail map. This is also done when the trackball moves.

## 5 Conclusions

We believe that a 3D shaded display can add a significant amount of information to the visualization of high-dimensional clusters, and that it allows users to gain better insight into higher dimensional relationships in the data. This is because a 3D shaded display makes better use of the innate human capability to recognize and reason with 3D information. Our study of a variety of real-world datasets showed that the extension from the traditional 2D space to 3D space is indeed justified – most datasets we studied had clusters residing in subspaces with more than two significant principal components. With this justification in hand, we designed an interactive interface that allows users to navigate these 3D subspaces, expand the exploration to higher dimensionalities, and also transition among the distinct subspaces inhabited by different clusters in the data.

For the future, we like to enhance our tessellation and graphics engine. Currently, the solid enveloping the point cloud of a cluster has a rather low resolution and is convex. We would like to refine the tessellation routine such it can better reproduce fine detail and also capture concave shapes and concavities within a shape. Further capabilities we would like to add is the ability to intersect (partially) overlapping solids and render hierarchical representations in which subdividing clusters are rendered within the solid of its parent cluster.

Another rendering paradigm, alternative to the current polygonal graphics, is volume rendering. It would enable a better rendition of the appearance properties of the point cloud, such as skew, varying density, and outliers. Each point could be represented as a radial basis function and rendering complexity could be handled by a level of detail solution as well as GPU-acceleration. If this proofs too slow for practical application, then texturing the surface with projections of the point cloud could be another feasible solution, invoking concepts from image-based rendering.

Finally, we would like to expand our study of datasets and also conduct extensive user studies to refine our interface and test its application scope and relevance.

## References

[1] D. Asimov, "The Grand Tour: A Tool for Viewing Multidimensional Data," *SIAM J. Scientific and Statistical Comp.*, 6(1):128-143, 1985

[2] E. Angel, D. Shreiner, *Interactive Computer Graphic with WebGL* (7th Edition), Addisoin-Wesley, 2014.

[3] N. Elmqvist, P. Dragicevic, J.-D. Fekete, "Rolling the dice: multidimensional visual exploration using scatterplot matrix navigation," *IEEE Trans. Visualization and Computer Graphics*, 14(6):1539-1148, 2008.

[4] M. Ester, H. Kriegel, J. Sander, X. Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise." *KDD*, vol. 96, pp. 226-231. 1996.

[5] J. Hartigan, "Printer graphics for clustering," *Journal of Statistical Computation and Simulation*, 4(3):187-213, 1975.

[6] G. Hatfield, "Perception as Unconscious Inference," *Perception and the Physical World: Psychological and Philosophical Issues in Perception*, D. Heyer and R. Mausfeld, eds., pp. 115-143, Wiley, 2002.

[7] D. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang,"iPCA: An Interactive System for PCA-Based Visual Analytics," Computer Graphics Forum, 28(3):767-774, 2009.

[8] K. Kailing, H. Kriegel, P. Kröger. "Density-connected subspace clustering for high-dimensional data." *Proc. SDM*, 4, 2004.

[9] H. Kriegel, P. Kröger, A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering." *ACM Trans. on Knowledge Discovery from Data,* 3(1):1, 2009.

[10] J. Nam, K. Mueller, "TripAdvisorN-D: A Tourism-Inspired High-Dimensional Space Exploration Framework with Overview and Detail," *IEEE Transactions on Visualization and Computer Graphics,* 19(2): 291-305, 2013.

[11] M. Sedlmair, T. Munzner, M. Tory. "Empirical guidance on scatterplot and dimension reduction technique choices." *IEEE Trans. on Visualization and Computer Graphics*, 19(12): 2634-2643, 2013.

[12] D. Swayne, D. Lang, A. Buja, D. Cook, "GGobi: Evolving from XGobi into an extensible framework for interactive data visualization," *Comp. Statistics & Data Analysis,* 43(4):423-444, 2003.

[13] J. Tenenbaum, V. De Silva, J. Langford. "A global geometric framework for nonlinear dimensionality reduction." *Science*, (290) 5500: 2319-2323, 2000.

[14] A. Tatu,, L. Zhang, E. Bertini, T. Schreck, D. Keim, S. Bremm, T. von Landesberger. "Clustnails: Visual analysis of subspace clusters." *Tsinghua Science and Technology,* 17(4): 419-428, 2012.

[15] S. Verheyen, E. Ameel, G. Storms. "Determining the dimensionality in spatial representations of semantic concepts," *Behavior Research Methods*, 39(3): 427-438, 2007.

[16] X. Yuan, D. Ren, Z, Wang, C. Guo, "Dimension projection matrix/tree: interactive subspace visual exploration and analysis of high dimensional data," *IEEE Trans. on Visualization and Computer Graphics,* 19(12):2625-2633, 2013.

[17] Z. Zhang, X, Tong, K. McDonnell, A. Zelenyuk, D. Imre, K. Mueller. "An interactive visual analytics framework for multi-field data in a geo-spatial context." *Tsinghua Science and Technology* 18(2): 111-124, 2013.

[18] https://archive.ics.uci.edu/ml/datasets/Housing

[19] https://archive.ics.uci.edu/ml/datasets/Image+Segmentation