Finding and Visualizing Relevant Subspaces for Clustering High-Dimensional Astronomical Data Using Connected Morphological Operators

Bilkis J. Ferdosi[1]* Hugo Buddelmeijer[2]† Scott Trager[2]‡ Michael H.F. Wilkinson[1]§
Jos B.T.M. Roerdink[1]¶

[1]Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen
[2]Kapteyn Astronomical Institute, University of Groningen

Abstract

Data sets in astronomy are growing to enormous sizes. Modern astronomical surveys provide not only image data but also catalogues of millions of objects (stars, galaxies), each object with hundreds of associated parameters. Exploration of this very high-dimensional data space poses a huge challenge. Subspace clustering is one among several approaches which have been proposed for this purpose in recent years. However, many clustering algorithms require the user to set a large number of parameters without any guidelines. Some methods also do not provide a concise summary of the datasets, or, if they do, they lack additional important information such as the number of clusters present or the significance of the clusters. In this paper, we propose a method for ranking subspaces for clustering which overcomes many of the above limitations. First we carry out a transformation from parametric space to discrete image space where the data are represented by a grid-based density field. Then we apply so-called connected morphological operators on this density field of astronomical objects that provides visual support for the analysis of the important subspaces. Clusters in subspaces correspond to high-intensity regions in the density image. The importance of a cluster is measured by a new quality criterion based on the dynamics of local maxima of the density. Connected operators are able to extract such regions with an indication of the number of clusters present. The subspaces are visualized during computation of the quality measure, so that the user can interact with the system to improve the results. In the result stage, we use three visualization toolkits linked within a graphical user interface so that the user can perform an in-depth exploration of the ranked subspaces. Evaluation based on synthetic as well as real astronomical datasets demonstrates the power of the new method. We recover various known astronomical relations directly from the data with little or no *a priori* assumptions. Hence, our method holds good prospects for discovering new relations as well.

Keywords: Subspace finding, clustering high-dimensional data, connected morphological operators, visual exploration, astronomical data.

Index Terms: H.3.3 [Information Search and Retrieval]: Clustering; J.2 [Computer Applications]: Physical Sciences and Engineering—Astronomy; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1 Introduction

Data sets in astronomy are growing to enormous sizes. Modern astronomical surveys provide not only image data but also catalogues of millions of objects (stars, galaxies), each object with hundreds of associated parameters. Although the high data rates required for acquisition, processing and populating the archives are well under control and are supported by dedicated projects teams, we need to develop new approaches for extracting, analyzing and visualizing astronomically relevant information out of the flood of data. Exploration of very high-dimensional information spaces poses a huge challenge. On the one hand, the techniques should cope with enormous amounts of data in a highly automated fashion, and be scalable to ensure that the newly developed methods remain usable while the data catalogues increase in size. On the other hand, the approach should allow the observer to participate in the analysis by using interactive visualization combined with the human perceptive and analytical power. This is especially true as the goal is to find "unexpected" phenomena in the data, for which by definition no *a priori* description is available, thus precluding the possibility of fully automated search.

Combining data mining approaches with visualization can enable users to explore such large datasets. Clustering is a well known data mining task that helps to discover natural structures in a dataset [20]. Due to the exploratory nature of the task, full dimensional clustering techniques cannot help much. Clusters may exist in different subspaces that may indicate different relations among particular subsets of dimensions. Subspace clustering is an approach that can be applied for this purpose. Subspace clustering is the process of finding clusters in subspaces of the full feature space, either directly [4] or by identifying relevant subspaces for (later) clustering based on some quality criteria [9].

In this paper we propose an approach to find relevant subspaces which is strongly tied to morphological properties of object distributions. Therefore, we apply techniques from the field of mathematical morphology, which was developed to describe image operators for enhancement, segmentation and extraction of shape information from digital images [17, 27].

The main steps of our approach can be summarized as follows. First we carry out a transformation from the parametric space of astronomical objects (stars, galaxies) to a discrete image space where the data are represented by a density field. This transformation is obtained by using grid based-density estimation. Next we determine for each local maximum of the density field whether it represents a relevant subspace by applying quality criteria based upon the notion of *dynamics* [10], which indicates the significance of a local maximum, see section 3.4.

The search for modes/local maxima is done on the so-called *Max-tree representation* of the density image. Such a representation is used in mathematical morphology to implement an important class of morphological operations known as *connected opera-*

*e-mail: b.j.ferdosi@rug.nl
†e-mail:buddel@astro.rug.nl
‡e-mail:sctrager@astro.rug.nl
§e-mail:m.h.f.wilkinson@rug.nl
¶e-mail:j.b.t.m.roerdink@rug.nl

tors [24, 26]. The main property of connected operators is that they do not process individual data points, but entire connected components at each grey level. Such components are either kept or completely removed by the operator. Therefore, such operators can be used to perform filtering based on various shape and size attributes. More information on connected operators is provided in section 3.3. For subspaces of dimension higher than three we apply principal component analysis (PCA) and use the first three principal components for subspace ranking. The main reason for using PCA is that for higher dimensions the current Max-tree implementation becomes prohibitive in terms of computing time and memory use.

Along with the quality measure and ranking of the subspaces we provide quantitative information such as the number of clusters present, degree of separation, size and shape of the clusters, etc. Note that our method does not perform the actual clustering itself, i.e., it does not assign points to clusters. For this purpose, existing clustering algorithms (such as k-means) may be used.

Visualization plays an important role in our approach. The subspaces are visualized during computation of the quality measure, so that the user can interact with the system to improve the results. In the result stage, we use an interactive tree visualization providing all sorts of statistics about each subspace along with the ranking. We also link three visualization toolkits within a graphical user interface so that the user can perform an in-depth exploration of the ranked subspaces.

Our main contributions can be summarized as follows:

- We introduce the use of connected morphological operators to analyze grid density profiles of subspaces of parameter space;
- We propose a subspace quality criterion based on the dynamics of maxima found in the density profile;
- Linked visualizations are used to support the user in the exploration of the subspaces.

The remainder of the paper is organized as follows. Related work is discussed in section 2. Section 3 then describes the working principle of our subspace finding method, including the background on density estimation, connected morphological operators and the concept of dynamics. Our interactive visual subspace exploration system is described in section 4. We present the experimental results of the method in section 5. Section 6 gives a summary along with plans for future work.

2 RELATED WORK

A well known method to rank subspaces for clustering is the SURFING ("SUbspace Relevant For clusterING") method [9]. It belongs to the class of methods that only compute interesting subspaces rather than final subspace clusters [20]. Relevance of a (sub)space is measured through a quality criterion based on a hierarchical clustering structure of subspaces. The method is based on the idea that subspaces with clusters of different densities and noise will show significant variation in k-nearest neighbor distances compared to subspaces with a uniform distribution. The quality of a subspace is determined as a function of differences of distances to the mean distance of the objects. SURFING can be very helpful where in-depth knowledge of the spaces can be traded against high processing speed, e.g., in web services. However, this method only gives a qualitative ranking of the subspaces without any quantitative information such as the number, size, shape or separation of the clusters.

There are other methods like CLIQUE (CLustering In QUEst) [4], ENCLUS (ENtropy-based CLUStering) [13], DOC (Density-based Optimal projective Clustering) [23], or PROCLUS (PROjected CLUStering) [3] that perform direct cluster computation in subspaces. CLIQUE first finds candidate subspaces by computing a histogram in each of the dimensions and selecting the dense ones. Then clusters are computed in the subspaces that are selected by a criterion that satisfies a downward closure (or monotonicity) property [20]. Pruning subspaces is done by the MDL (Minimal Description Length) principle. However, CLIQUE provides no information on the subspaces in which the whole dataset clusters best. Top-down pruning can miss many small but significant clusters. It also is difficult to find a proper tuning of parameters for different datasets.

Integration of visualization in the subspace ranking and clustering process seems to be a less explored area. Assent et al. [6] proposed a visualization paradigm to present and explore clusters from subspace clustering. Using multidimensional scaling (MDS) they present information like (dis)similarity, overlap, size, dimensionality etc., of the resulting clusters. They provide an aid to parameter tuning in terms of *bracketing*, a technique originating from photography. A matrix representation is used to visualize the grouping of clusters. However, these visualization approaches are about presentation of clustering results, but do not aid in exploring individual subspaces, our goal in this paper.

3 SEARCHING RELEVANT SUBSPACES FOR CLUSTERING

3.1 Overview of the method

Let us denote by $DATA$ a set of N data points (rows) with d dimensions (columns), i.e., $DATA \subseteq \mathbb{R}^d$. Let $A = \{a_1, ..., a_d\}$ be the set of all attributes a_i of $DATA$. A subspace in $DATA$ is a set S with $S \subseteq A$. We define a subspace as *relevant* if it does not contain uniform noise or only a single Gaussian distribution spread over the whole attribute range. Therefore, the emphasis is given on multimodality of the density where each mode is indicative of a cluster. The degree of relevance is determined in terms of significance and separability of each mode (indicator of a cluster) in the multimodal distribution.

We search for the modes and determine their significance and separability in grey level image space, whereas most of the traditional subspace clustering methods work in parametric space. The motivation for working in discrete image space is that the number of grid points can be chosen to match the desired grid resolution, while the number of data points may grow very large. This representation facilitates the analysis of the subspaces because of the structured representation using the Max-tree. Also, it allows an easy integration of the visualization of the density field.

Therefore, a transformation of parametric space to image space is required. This transformation is obtained by using grid-based density estimation, as described in section 3.2. Thus modes in the distribution are transformed into high-intensity peaks (local maxima) in the density image.

The search for modes/local maxima is done on the Max-tree representation of the density image, see section 3.3. Each node of the Max-tree with a certain grey level contains all the connected components at that level. Connected components are obtained using neighborhood relationships in the grid. The root of the tree contains the connected components with lowest intensity and the leaves contain the connected components with highest intensity. Therefore, counting the number of leaves gives us the number of clusters.

The significance and separability of modes is determined using the concept of relative dynamics as described in section 3.4. Significant and well-separated modes will have higher relative dynamics compared to overlapping clusters. To derive a quality criterion for subspaces we use the number of modes (number of leaves in the Max-tree) and their relative dynamics, see section 3.5.

3.2 Density estimation

Density estimation is one of the techniques of choice to uncover structure in point-set data [28]. We estimate the density of each subspace by a fast and scalable modification from [30] of the adaptive kernel density estimation method of Breiman *et al.* [11].

For a data sample of N points with position vectors $\vec{r}_i = (r_{1,i}, r_{2,i}, \ldots, r_{d,i}) \in \mathbb{R}^d, (i = 1, ..., N)$, the adaptive kernel density estimate $\hat{p}(\vec{r})$ is given by:

$$\hat{p}(\vec{r}) = \frac{1}{N}\sum_{i=1}^{N}(h_1 \ldots h_d)^{-1}\lambda_i^{-d} K_e\left(\frac{r_1 - r_{1,i}}{h_1\lambda_i}, \ldots, \frac{r_d - r_{d,i}}{h_d\lambda_i}\right) \tag{1}$$

Here the local bandwidth is the product of a window size h_ℓ depending on the coordinate direction $\ell = 1, 2 \ldots, d$ and a local bandwidth parameter λ_i for each data point i. In this formula K_e is the Epanechnikov kernel defined as

$$K_e(\vec{t}) = \begin{cases} \frac{d+2}{2V_d}(1 - \vec{t}.\vec{t}) & \text{if } \vec{t}.\vec{t} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

in which V_d is the volume of the unit sphere in d-dimensional space. In this model we have to choose the local bandwidth parameters λ_i in such a way that in low-density regions λ_i will be large and the kernel will spread out; in high-density regions the opposite should occur.

The density estimation proceeds in two phases.

Phase 1. Use a percentile of the data to compute an optimal pilot window width h_ℓ^{opt} in each of the coordinate directions:

$$h_\ell^{opt} = \frac{P_{80}(\ell) - P_{20}(\ell)}{\log N}, \quad \ell = 1, \ldots, d \tag{3}$$

where $P_{80}(\ell)$ and $P_{20}(\ell)$ are the 80th and 20th percentile of the data points in dimension ℓ. Define a pilot density $\hat{p}_{pilot}$ using $\lambda_i = 1$ for all $i = 1, 2 \ldots, N$ and $h_\ell = h_\ell^{opt}$ in formula (1).

Phase 2. From the pilot density $\hat{p}_{pilot}$ compute the local bandwidth parameters λ_i by

$$\lambda_i = \left(\frac{\hat{p}_{pilot}(\vec{r}_i)}{g}\right)^{-\alpha}. \tag{4}$$

Here g is the geometric mean of the pilot densities and $\alpha = 1/d$ is the sensitivity parameter. The final density estimate is given by formula (1) once again, but now with λ_i given by (4) with $h_\ell = h_\ell^{opt}$.

The Epanechnikov kernel has finite support so that computation time is reduced significantly. The density is estimated on a Cartesian grid, which includes all data points. The method is computationally effective: the complexity is $O(N)$; the computation time will increase for larger values of the smoothing parameter. Because of its grid structure the computed density can be visualized immediately by standard volume rendering techniques for $d \leq 3$. In our method a fundamental use of the grid structure is to obtain a neighborhood definition for computing connected components in the density field. Note that the grid must be finer than the smallest window size.

3.3 Connected morphological operators

A connected operator can extract and filter connected components known as flat zones, i.e., constant intensity regions, where connectivity is defined on the digital grid. Connected operators create a hierarchy of flat-zone partitions with an ordering relation. The Max-tree data structure can be used to implement such a hierarchy [24, 25].

Consider a digital image I on a domain $D \subseteq \mathbb{Z}_n$ with 2-adjacency for n = 1, 4 or 8-adjacency for $n = 2$, and 6 or 26-adjacency for $n = 3$. A set $X \subseteq D$ is connected if each pair (p, q) of points in X can be joined by a path $(p_0, p_1, \ldots, p_{\ell-1}, p_\ell)$ such that $p_0 = p$, $p_\ell = q$ and (p_i, p_{i+1}) are neighbors $\forall i \in [0, \ell)$. A connected component of X is a connected subset $C(X)$ of X which is maximal. A flat zone at grey level h of I is a connected component of the level set $X_h(I) = \{p \in D | I(p) = h\}$.

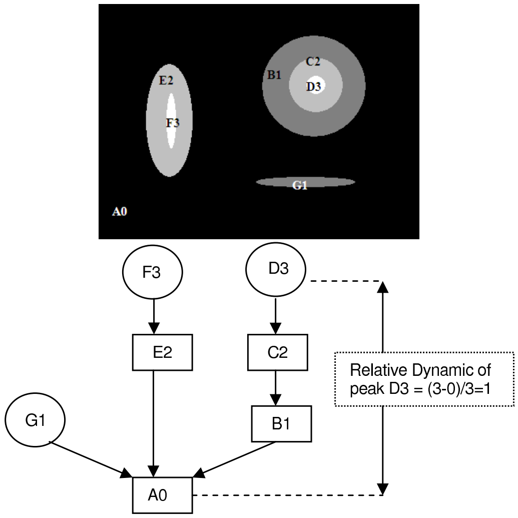


Figure 1: Top: grey level image that contains three connected components with varying intensity. Bottom: Max-tree representation of the top image. Max-tree node A0 represents the background, and the other connected components are indicated by B to G along with their grey values. The relative dynamics of peak D3 is also indicated.

Max-tree representation. In the Max-tree representation of an image the root corresponds to the flat zone with lowest intensity and leaves contain the flat zones with highest intensity [24, 25]. Local maxima in the image correspond to connected sets of constant value which are separated from other local maxima by local minima. An illustration is given in Fig. 1. In the top image of this figure there are three well-separated clusters with varying intensity. In the bottom image the corresponding Max-tree representation is shown. The Max-tree node A0 represents the background. As there are two flat zones with grey level 1 and one with grey level 2, the root has two child nodes (B1, G1) at level 1 and one child node (E2) at level 2. Each of the flat zones can be a leaf or have children. Flat zones with maximum intensities are in the leaves (G1, F3, D3). The Max-tree is a rooted tree, thus every node has a pointer to its parent. The Max-tree is constructed with a recursive flood filling with a FIFO queue to process the pixels/voxels in the correct order.

Each node in the Max-tree can contain several size or shape attributes that can be calculated incrementally during the tree construction. Some example attributes are *Size*, i.e., the area A of the flat zone as defined by the number of pixels in that zone, or the scale invariant shape attribute defined by M/A^2, i.e., the ratio of moment of inertia M and the square of the area A. The Max-tree along with the attributes can be computed in a time which is linear in the number of pixels.

3.4 Dynamics

In image analysis the concept of “dynamics” is used as a measure of contrast. It can be used to rank the local maxima of an image [10]. The dynamics of a local maximum is defined as the difference between the height H_1 of that maximum and the height H_2 of the deepest neighboring minimum (as shown in Fig. 2), i.e., $Dynamics(m)$ = $H_1 - H_2$. The computation of the dynamics of local maxima becomes easy in the Max-tree structure of the image. In the Max-tree the local maxima are in the leaves. Therefore, the dynamics of a local maximum is the difference between the intensity value of the corresponding leaf and the intensity value of the first ancestor with multiple children that corresponds to the deepest minimum in the neighborhood (cf. Fig. 1). One problem with this definition is that a maximum with low amplitude can be treated as insignificant compared to a maximum with large amplitude. Therefore, we use relative dynamics so that all maxima are treated equally, i.e., when

m is a local maximum its relative dynamics is defined by

$$RelativeDynamics(m) = (H_1 - H_2)/H_1. \tag{5}$$

For the example of Fig. 1 this means that all the maxima have a relative dynamics of 1. Relative dynamics are also scale-invariant, because a linear scaling of the data space scales all the densities linearly as well, thus the relative differences in density between extrema and saddle points remain the same.

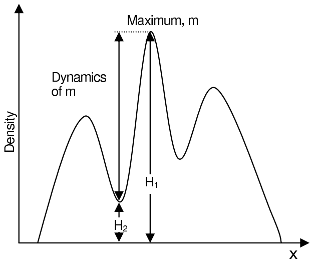


Figure 2: Dynamics of local maximum m.

3.5 Subspace quality criterion

Let S be a subspace of the space A of attributes. The quality of S, denoted by *Quality*(S), is defined as follows

$$Quality(S) = \begin{cases} N_L^{-1} \sum_{i=1}^{N_L} RelativeDynamics(i) & \text{if } N_L > 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where N_L is the number of leaves in the Max-tree. In this criterion the sum of the dynamics of all local maxima is normalized by the number of local maxima and thus the value of *Quality* ranges from 0 to 1. A subspace that contains modes/clusters with high dynamics will have a higher value of *Quality* than a subspace with clusters of lower dynamics. A subspace as depicted in Fig. 1 will have a quality of 1 according to equation (6) because of the presence of three modes with dynamics of 1 each. Two important aspects of our quality criterion are: (*i*) the use of relative dynamics allows us to treat clusters with varying density equally; (*ii*) the quality criterion is unbiased in ranking subspaces with varying number of clusters because of the normalization by the number of leaves.

Note that in our method, subspaces with the same quality, but with varying numbers of clusters, get the same ranking and thus they will be grouped together in the rank list. However, along with the ranking, our method also provides information about the number of clusters that may be present. Therefore, it becomes possible for the user to choose the subspace of interest (with more/less clusters) from the group of subspaces with the same quality, unlike other methods where such grouping is not available.

3.6 Subspace finding

The search for subspaces is performed in a bottom-up fashion, i.e., starting from one-dimensional subspaces, then moving to two-dimensional subspaces, etc. The process of finding relevant subspaces is summarized in the pseudocode of Algorithm 1. Up to dimension three the creation of the density image, Max-tree construction and computation of the quality index is done on the original feature space. For subspaces of dimension higher than three we apply PCA and use the first three principal components for subspace ranking. The main reason is that for higher dimensions the current Max-tree implementation becomes prohibitive in terms of computing time and memory use. Using PCA globally in the full dimensional feature space is open to criticism. However, in our approach we are using it in local feature spaces. Therefore, we can avoid the drawbacks of global usage of PCA. An added benefit of our choice to use the first three principal components of PCA is that we can use standard volume rendering to visualize the density fields.

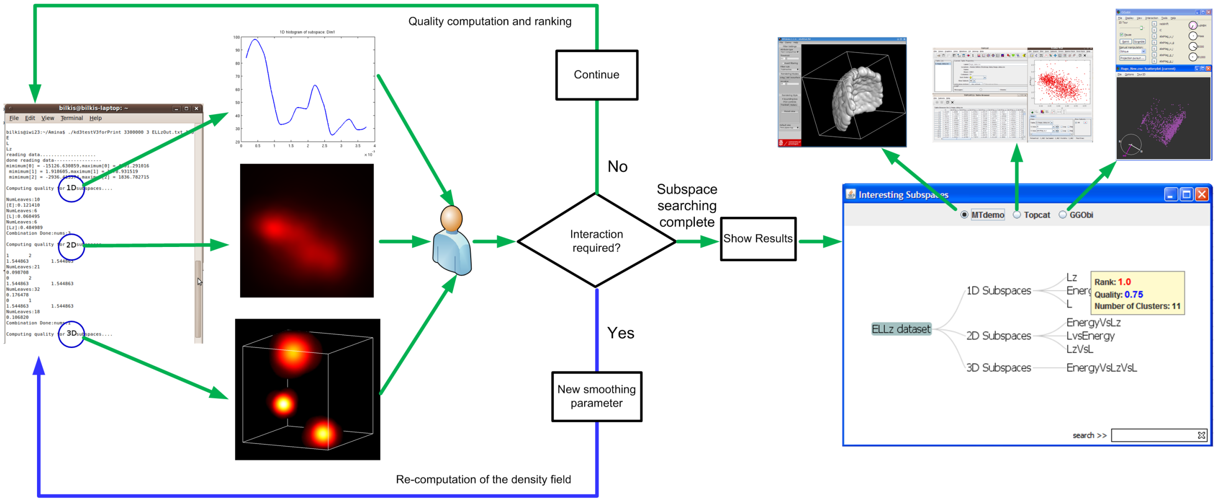
Ranking and Pruning. Based on the quality of the subspaces we produce a ranking. Unlike SURFING we do not discard any of the subspaces in the one dimensional search. Discarding spaces in such an early stage can reduce the search space dramatically but it also precludes the possibility of finding interesting relations in later stages that may arise with the combination of discarded 1-D subspaces. However, it is necessary to prune the subspaces because of their exponential growth. Therefore, we introduce pruning for 2-D and higher dimensions. We prune a subspace if it has a quality value less than a threshold value θ. From our study on several uniformly distributed spaces we found that they always have a quality value less than 0.1. Therefore, we set $\theta = 0.1$.

Algorithm 1 SubspaceFinding

```
 1: $DATA \leftarrow d$-dimensional dataset;
 2: $A=\{a_1.,..,a_d\}$; // attribute set
 3: $n = 1$;
 4: while $n \leq d$ do
 5:     $NrOfSpaces \leftarrow \binom{d}{n}$;
 6:     $S_n \leftarrow$ set of $n$-dimensional subspaces $S_{n,j}$;
 7:     for $j = 1$ to $NrOfSpaces$ do
 8:         if $(n > 3)$ then
 9:             $S_{n,j} \leftarrow$ ComputePCA($S_{n,j}$);
10:         end if
11:         $Den_{n,j} \leftarrow$ ComputeDensityField($S_{n,j}$);
12:         Visualize($Den_{n,j}$);
13:         WaitForInteraction;
14:         if ($interaction$) then
15:             Accept new smoothing parameter
16:             go to 11;
17:         else
18:             $M_{n,j} \leftarrow$ CreateMaxTree($Den_{n,j}$);
19:             quality($S_{n,j}$) $\leftarrow$ ComputeQuality($M_{n,j}$);
20:         end if
21:     end for
22:     $rank$ according to quality;
23:     //Pruning for $n > 1$
24:     if ($n > 1$ and quality$(S_{n,j}) < \theta$) then
25:         remove $S_{n,j}$;
26:     end if
27:     $n \leftarrow n + 1$;
28: end while
```

4 Interactive visual subspace exploration

An overview of our subspace search and exploration system is given in Fig. 3. The left part of the figure shows the quality computation process. It is very important to choose a proper value for the smoothing parameter during density computation. Most of the current density-based approaches for subspace clustering and ranking try to find a proper parameter by trial and error, which is very cumbersome [21]. Initially, we provide an automatic setting of the smoothing parameter as described in section 3.2. Most of the time this automatic selection works. However, it may produce an under / over-smoothed density field, which is best detected through visual inspection by the user. Therefore, in our method we visualize the density field with standard volume visualization for 3-D and higher dimensions. For 2-D we visualize it as an image and for 1-D the histogram of the point densities is used. If the user detects any over / undersmoothing s/he can interact with the system to give a new smoothing parameter value.



```
bilkis@iwi23:~/Amina$ ./kd3testV3forPrint 3300000 3 ELLzOut.txt
E
L
Lz
reading data....................
done reading data----------------
mimimum[0] = -15126.630859,maximum[0] = 4091.291016
 mimimum[1] = 1.918605,maximum[1] = 1778.931519
 mimimum[2] = -2936.413574,maximum[2] = 1836.782715

Computing quality for subspaces....

NumLeaves:10
[E]:0.121410
NumLeaves:6
[L]:0.060495
NumLeaves:6
[Lz]:0.484989
Combination Done:nums:3

Computing quality for subspaces....

1       2
1.544863        1.544863
NumLeaves:21
0.098708
0       2
1.544863        1.544863
NumLeaves:32
0.176478
0       1
1.544863        1.544863
NumLeaves:18
0.106820
Combination Done:nums:1

Computing quality for subspaces....
```

Rank: 1.0
Quality: 0.75
Number of Clusters: 11

Figure 3: Schematic diagram of our interactive search and exploration system.

We represent the result of the relevant subspace finding method by a tree visualization (see right side of Fig. 3). The root represents the complete dataset, the next level contains d nodes where d is the dimensionality of the dataset. Each node contains a number of leaves, say m, where m is the number of relevant subspaces. If the mouse pointer hovers over a node an information box will appear with all the relevant information about that subspace. By clicking on the node a window will appear with a 1-D or 2-D density plot for dimension one and two, and a volume visualization of the density field for dimension three. For dimensions higher than three the density field of the first three principal components is visualized.

The tree can be panned (scrolled) to explore the branches. The user can also zoom in/out for better reading in case that the tree is large or too cluttered. We combine three visualization tools with our interface. From the top panel the user can choose Topcat[1], GGobi[2] or MTdemo[3] to check if the subspaces are really relevant. Topcat is a well known table visualizer in the astronomical community that also has different plotting capabilities. It is quite competent in handling very large high-dimensional data. GGobi is also a well-known information visualization toolkit that provides several high-dimensional data visualization techniques. For volume visualization we use MTdemo, a Max-tree-based volume visualization tool presented by Westenberg et al. [29]. It renders the volume with three different rendering techniques, X-ray, Maximum Intensity Projection (MIP) and Isosurface. MTdemo is not only a volume visualization tool but also an attribute filtering tool. It allows the user to explore the volume by applying different shape preserving attribute filters.

The amount of interaction will differ in the two phases. In the subspace ranking phase the smoothing parameter can be changed interactively. To make the subspaces comparable we normalized the units along each axis. Therefore, the scaling parameter should not vary for different subspaces of a particular dimension. Thus, inspecting one subspace per dimension should be sufficient. Still the number of inspections per dimension will depend on how often the smoothing parameter is changed, which can vary from user to user. Once the subspace ranking is complete, the number of inspections will be limited, as the subspaces are ranked by relevance. Usually, domain users have concrete hypotheses they want to verify and hence they will only explore the most relevant subspaces.

5 Experiments and Results

We compare the ranking performance of our method with SURFING, and the performance in finding the number of clusters with CLIQUE, as SURFING does not provide the latter information. As the source code was not available to us we used our own implementation of SURFING following the algorithm presented in [9]. For CLIQUE we used the ELKI[4] platform [2]. We used several synthetic datasets and two astronomical datasets for this purpose. Reported timings were obtained with an AMD athlon 64 X2 Dual core processor 5200+, 2.6 GHz and memory 1.94 GB.

5.1 Synthetic datasets

We created several synthetic datasets with varying numbers of clusters of varying dimensionality with different noise levels in a uniform box of size 100. Clusters were created as multimodal Gaussian distributions with different mean and variance. Depending on the value of the variance we created clusters with varying density. Then impulse noise was inserted uniformly, where the number of noise points varied from 0% to 10% of the number of points in the clusters. In Table 1 a brief summary of the synthetic datasets can be found. The field "data dimension" indicates the dimensionality of the dataset. "Number of clusters" indicates the number of Gaussian clusters present in the dataset and "Cluster dimensions" indicates the dimensionality of the Gaussian clusters. For example in dataset 2, the dimensionality of the dataset is 12 $(d_1, d_2, \ldots, d_{12})$, and there are four 3D Gaussian clusters (in d_2, d_4, and d_6) with 10% uniformly distributed noise added and two 6D Gaussian clusters (in $d_7 - d_{12}$) present in the datasets without noise; the remaining dimensions (d_1, d_3, and d_5) of the dataset contain uniformly distributed random noise.

Table 1: Synthetic datasets

Dataset	Data Dimension	Number of clusters	Cluster dimensions
1	16	2	3
2	12	4,2	3,6
3	15	3	4
4	22	5	5
5	12	3	2

[1] http://www.star.bris.ac.uk/~mbt/topcat

[2] http://www.ggobi.org

[3] http://www.cs.rug.nl/~michael/MTdemo

[4] http://www.dbs.ifi.lmu.de/research/KDD/ELKI

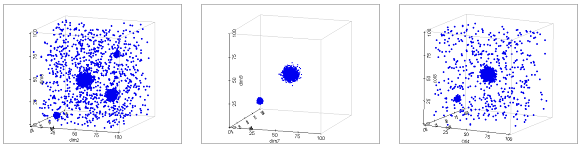


Figure 4: Scatter plot of subspace-A from dataset 2 (left), subspace-B from dataset 1 (middle), subspace-C from dataset 1 with noise added (right).

Table 2: Comparison of methods on *Galactic stellar halo* dataset

Method	Ranking		Nr. clusters indicated
	1-D	2-D	
Our method	L_z E L	$E-L_z$ $E-L$ $L-L_z$	31
SURFING	E L L_z	$L-L_z$ $E-L_z$ $E-L$	n.a.
CLIQUE	(in terms of coverage) E L L_z	(in terms of coverage) $L-L_z$ $E-L_z$ -	15

Performance for synthetic data. The performance of our method for synthetic datasets is satisfactory. It ranks subspaces with clusters always high in the list irrespective of the noise levels. It ranks subspace-A,B,C (Fig. 4) as equally relevant since they all get a quality value of 1. Our method puts emphasis not only on the number of clusters but also on their separability. Subspaces that have well separated clusters always come up high in our ranking. It also can indicate the number of clusters properly in most of the cases. Sometimes fewer clusters than present are reported if there are overlapping clusters with one very high density and another with very low density.

SURFING puts most of the subspaces which do contain clusters higher in the ranking. However, noise-free cluster structures are penalized compared to clusters with noise in this method, see 'subspace-A' and 'subspace-B' in Fig. 4, left and middle, respectively. In subspace-A there are four Gaussian clusters of varying density with uniformly distributed noise that covers all clusters. In subspace-B there are two clusters without noise. The SURFING method put subspace-A in the top ranking as expected. However, it ranked subspace-B only as the 20th relevant subspace in the list. Note that this result was obtained in spite of the fact that we introduced 1% of additional random points when calculating the SURFING quality measure, as recommended by Baumgartner *et al.* [9]. The motivation for adding a small percentage of random points is that SURFING's quality measure is based on the difference between k-nearest neighbor distances and mean distances. Hence, if a subspace has multiple clusters with the same density and without noise, it would get the same quality value as uniformly distributed points and thus remain lower in the ranking. By contrast, for cases where the clusters are fully covered by noise, as in Fig. 4 right, we found that SURFING does rank the subspace equally high as subspace-A in the list of relevant subspaces.

CLIQUE missed some of the clusters and sometimes detected unclear clusters. The main difficulty of CLIQUE is the need to find proper parameter sets that work for individual datasets.

For the synthetic datasets we checked whether the use of PCA caused any high (i.e., larger than 3) dimensional clusters to be missed. We found that this only occurred in dataset 3, where one of the three 4D clusters was missed. In dataset 4 all the five 5D clusters and in dataset 2 both 6D clusters were indicated.

5.2 Astronomical data

We used two astronomical datasets. The first one is the *Galactic stellar halo* (roughly spherical outskirts of a galaxy) dataset, which is the result of a simulation. The second is a galaxy sample from SDSS (*S*loan *D*igital *S*ky *S*urvey), cf. `http://www.sdss.org`.

Galactic stellar halo dataset. This consists of 33 satellite galaxies each of them represented by a collection of 10^5 particles. It has been assumed that the whole stellar halo is the superposition of several disrupted satellite galaxies which fell into the Milky Way about 10^{10} years ago. It is possible to isolate remnants of satellite galaxies since stars in galaxies harbour unique clues of the assembly history of galaxies. The dataset contains 33 satellites with three phase space parameters, i.e., energy E, total angular momentum L and the z-component of angular momentum L_z. These three parameters are approximately conserved quantities that do not evolve much. Among them only L_z is fully conserved and thus should play the most important role in finding clusters. According to Helmi and de Zeeuw [18] most structure is visible in the 2-D subspace $E-L_z$. However, they argued that all 33 clusters could be found in the $E-L-L_z$ space. With current approaches such as the *friends of friends* algorithm [15] only 50 percent of the clusters have been recovered so far.

We applied all the methods to the *Galactic stellar halo* dataset. The results are shown in Table 2. Our method has the best performance in correctly ranking the parameters and also in indicating the maximum number of clusters. The fact that our method is able to detect 31 out of 33 clusters is a great advance compared to the performance of state of the art astronomical methods which reach only half of this [18].

The ranking of our method is understandable if we look at the scatter plot of the 2-D spaces, see Fig. 5. The highest ranked 2-D subspace is $E-L_z$, which indeed has the largest number of clusters. However, CLIQUE's ranking in terms of coverage does not correspond to existing astronomical knowledge about the parameters. For example according to CLIQUE L_z has clusters with the least coverage of the dataset. However, according to Helmi and de Zeeuw L_z should contain more clustering information than the other parameters, as it is the most conserved quantity. Ranking of the 2-D subspaces is reasonable, although the method did not find any cluster in subspace $E-L$. CLIQUE found that subspace L-L_z has the clusters with highest coverage. It can be inferred that this subspace has less clusters with large size. CLIQUE found less than half of the clusters present.

The ranking of SURFING for this dataset corresponds to the results of CLIQUE. In 1-D subspaces energy E is in the top ranking, followed by L and L_z respectively. In 2-D subspaces L-L_z is indicated as the most relevant subspace. If we look at the scatter plot of Fig. 5 it is evident that the L-L_z and E-L subspaces have more variations in their point distribution in space. On the other hand, the E-L_z space has more clumped structures when compared to the other two subspaces. This may indicate the weakness of measuring relevance only based on variation in point distances.

Galaxy sample from SDSS. This data set contains mainly photometric information of galaxies in the Northern Galactic Cap of SDSS Data Release 7 [1]. There are 32228 galaxies with 15 attributes in total present in this dataset, see Table 3.

The sample is limited to a spectroscopically measured distance range of 418 to 460 Mpc (1Mpc $\approx 3*10^{19}$km) to control distance related effects. It is difficult to compare galaxies at different distances: they are observed at different cosmological times and with different recessional velocities. An upper r-band Petrosian [22] magnitude of 17.7 is imposed, to ensure a volume-complete sample for the quantification of the environment around the galaxies.

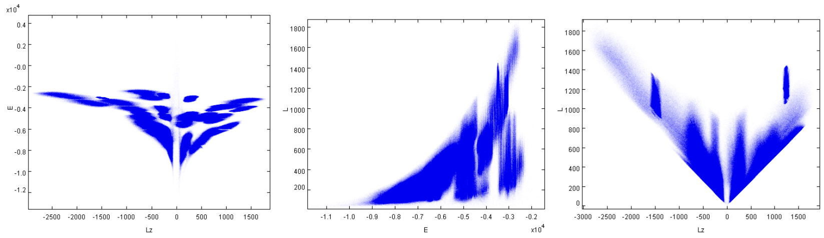


Figure 5: Scatter plot of *Galactic stellar halo* dataset. E vs L_z (left), E vs L (middle), L vs L_z (right).

Two of the attributes, i.e., *magnitude* and *color*, are important in optical astronomy and need some elaboration. Magnitude refers to the luminosity of a galaxy in a specific wavelength band of the electromagnetic spectrum. Higher magnitude values correspond to fainter objects, lower values to brighter objects. In our galaxy dataset we used extinction-corrected model magnitudes: *dered_r* is the magnitude of galaxies measured in the r-band (around a wavelength of 6280 Å). The colors of a galaxy are defined as the differences between magnitudes in two different bands [31] such that the higher the color value the redder the galaxies are. In this dataset 10 different colors are used, such as *u-r, u-g*, etc. This allows us to study the influence of different colors in finding galaxy properties. The (inverse) concentration index is a measure of the light distribution of a galaxy.

In our performance measurement on the *SDSS galaxy sample* dataset we recover several well known relations of galaxy properties. In color vs magnitude a bi-modal distribution of red and blue galaxies can be observed [8]. Red galaxies are elliptical galaxies with mostly old stars and blue galaxies are spiral galaxies with mostly young stars. In the color vs inverse concentration index relation, this galaxy bimodality can also be observed [7].

Table 3: Attributes used in *SDSS galaxy sample*

Attribute Name	Description
dered_r	Extinction corrected model-magnitude in the r-band.
10 colors: **u-g**, **u-r**, **u-i**, **u-z**, **g-r**, **g-i**, **g-z**, **r-i**, **r-z**, **i-z**	A quantitative measure of color of a galaxy is defined as the difference between magnitudes at two different effective wavelengths
logMass	Mass of the galaxy (in logarithmic scale)
logDensity	Number density of galaxies of the environment surrounding the galaxy (in logarithmic scale)
iC	Inverse Concentration index, a measure for the structure of the galaxy
SBr	Surface brightness of the galaxy

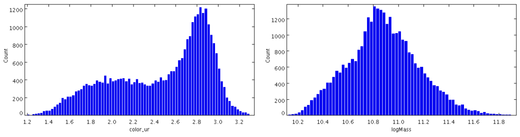


Figure 6: *SDSS galaxy sample* data set. Histograms of (left) *color(u-r)*: ranked 1 in our method, (right) *logMass*: ranked 1 in SURFING among 1-D subspaces.

In one dimension, the galaxy bimodality can be observed in the histogram of colors. Current astronomical research shows that this can best be seen in *color(u-r)*. This is confirmed by our method for ranking for 1-D subspaces, where *color(u-r)* is ranked first. On the other hand, SURFING ranked *logMass* highest. If we compare the histogram of these two subspaces (see Fig. 6) it is clear that *logMass* is not relevant in terms of clustering. On the other hand the *color(u-r)* histogram confirms the astronomical findings.

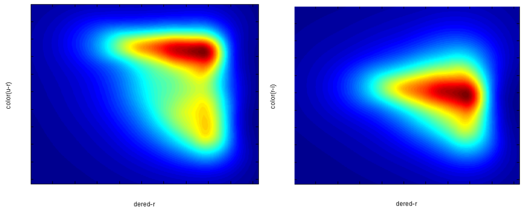


Figure 7: *SDSS galaxy sample* data set. Color vs Magnitude relation. Left: ranked 1 in our method: *dered_r* vs *color(u-r)*. Right: ranked 1 in SURFING: *dered_r* vs *color(r-i)*.

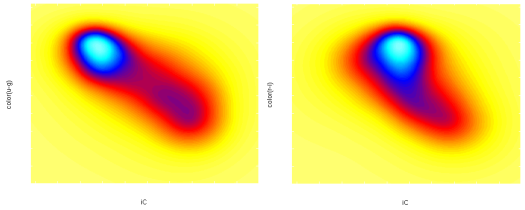


Figure 8: *SDSS galaxy sample* data set. Color vs inverse Concentration index relation. Left: ranked 1 in our method: *iC* vs *color(u-g)*. Right: ranked 1 in SURFING: *iC* vs *color(r-i)*.

When we search in 2-D subspaces the combination *dered_r* vs *color(u-r)* is the first subspace among color vs magnitude combinations that appears in the ranking of our method. On the other hand SURFING ranks *dered_r* vs *color(r-i)* first. We can see a clear bimodality in the density plot of *dered_r* vs *color(u-r)* subspace, see figure 7, whereas virtually no bimodality can be seen in the *dered_r* vs *color(r-i)* subspace. Similar observations hold for the *color* vs *iC* relation. Here we also found that the bimodality is best visible in the subspace chosen by our method, see Fig. 8. The performance of our

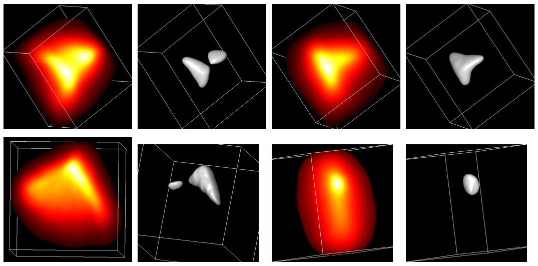


Figure 9: Visualization of *SDSS galaxy sample* dataset. Row 1: Volume visualization of 3-D subspaces. From left to right: ranked 1 in our method: *dered_r* vs *color(u-r)* vs *SBr* (Xray and isosurface); ranked 1 in SURFING: *dered_r* vs *color(i-z)* vs *SBr* (Xray and isosurface). Row 2: Volume visualization of first three principal components of 5-D subspaces. From left to right: ranked 1 in our method: *dered_r* vs *color(u-i)* vs *color(i-z)* vs *iC* vs *logMass* (Xray and isosurface); ranked 1 in SURFING: *color(g-r)* vs *color(g-z)* vs *color(r-i)* vs *color(i-z)* (Xray and isosurface).

method remains the same in higher dimensions, see Figure 9, where our method shows its strength in detecting relevant subspaces.

The performance of CLIQUE on the galaxy dataset is poor. We experimented with various parameter settings but could not find any of the known galaxy relations we were looking for.

Computation time. For synthetic dataset 1 (5000 data points) it took 0.001s, 1.15s, and 7.75s for computing the 1D, 2D, and 3D density field, respectively, while for the *Galactic stellar halo* dataset (with 3.3 million data points) it took 1.52s, 3.6s, and 217.72s (for both datasets with an automatic choice of the smoothing parameter).

6 Summary and future plans

In this paper we have presented a method for ranking subspaces in high-dimensional data in terms of their relevance for clustering. We used connected morphological operators on a grid-based density field that provides not only a good quality criterion but also has visual support for the analysis of the subspaces. Evaluation of the method on synthetic and astronomical datasets confirmed its strength in finding relevant subspaces and the usefulness of its visualization. In our approach we allow the user to interact with the system even during the search process, and directly confirm the results by looking into the density image produced. Our interactive application where tree visualization has been integrated with well-established visualization tools aids the user to achieve further in-depth knowledge by exploration of the subspaces.

Future work will concern further improvement of the results using dynamics-based filtering of the density image. We also will investigate extension of the Max-tree algorithm to dimension higher than three. This would enable subspace ranking without recourse to PCA in higher dimension. This however would also require the use of visualization techniques in dimension higher than three. Several methods are available for this purpose, such as parallel coordinate plots [19], scatter plot matrices [12], or tours [5, 14, 16].

The method will be extended for clustering subspaces. We will test the method on other astronomical datasets such as the RAVE survey, the Geneva-Copenhagen catalogue of nearby F and G stars and very large datasets provided by the OmegaCAM instrument. Also, application of the method on other domains like genomics or medical imaging will be considered. Finally, a user evaluation of the complete system is planned.

Acknowledgments

We thank prof. Amina Helmi of the Kapteyn Astronomical Institute for the Galactic stellar halo dataset used in this paper. This research is funded by the Dutch National Science Foundation (NWO), "STARE" program, project no. 643.200.501.

References

[1] K. N. Abazajian et al. The Seventh Data Release of the Sloan Digital Sky Survey. *Astrophysical Journal, Supplement*, 182:543–558, June 2009.

[2] E. Achtert, H. P. Kriegel, and A. Zimek. ELKI: A software system for evaluation of subspace clustering algorithms. In *SSDBM*, pages 580–585, 2008.

[3] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. *SIGMOD Rec.*, 28(2):61–72, 1999.

[4] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *ACM SIGMOD*, 27:94–105, 1998.

[5] A. Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM J. Sci. Statist. Comp.*, 6(1):128–143, 1985.

[6] I. Assent, R. Krieger, E. Müller, and T. Seidl. VISA: visual subspace clustering analysis. *SIGKDD Explorations*, 9(2):5–12, 2007.

[7] I. K. Baldry, M. L. Balogh, R. G. Bower, K. Glazebrook, R. C. Nichol, S. P. Bamford, and T. Budavari. Galaxy bimodality versus stellar mass and environment. *Monthly Notices of the Royal Astronomical Society*, 373:469–483, 2006.

[8] I. K. Baldry, K. Glazebrook, J. Brinkmann, Ž. Ivezić, R. H. Lupton, R. C. Nichol, and A. S. Szalay. Quantifying the bimodal color-magnitude distribution of galaxies. *Astrophysical Journal*, 600:681–694, Jan. 2004.

[9] C. Baumgartner, C. Plant, K. Kailing, H. Kriegel, and P. Kröger. Subspace selection for clustering high-dimensional data. In *In Proc. 4th IEEE Int. Conf. on Data Mining (ICDM'04*, pages 11–18, 2004.

[10] G. Bertrand. On the dynamics. *Image Vision Comput.*, 25(4):447–454, 2007.

[11] L. Breiman, W. Meisel, and E. Purcell. Variable kernel estimates of multivariate densities. *Technometrics*, 19:135–144, 1977.

[12] J. Chambers, W. Cleveland, B. Kleiner, and P. Tukey. *Graphical Methods for Data Analysis*. Wadsworth, 1983.

[13] C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93, New York, NY, USA, 1999. ACM.

[14] D. Cook et al. Grand tour and projection pursuit. *Journal of Computational and Graphical Statistics*, 4(3):155–172, 1995.

[15] G. Efstathiou, C. S. Frenk, S. D. M. White, and M. Davis. Gravitational clustering from scale-free initial conditions. *Monthly Notices RAS*, 235:715–748, Dec. 1988.

[16] E. D. Feigelson and G. J. Babu, editors. *Statistical Challenges in Astronomy*. Springer, Wien, New York, 2003.

[17] H. J. A. M. Heijmans. *Morphological Image Operators*, volume 25 of *Advances in Electronics and Electron Physics, Supplement*. Academic Press, New York, 1994.

[18] A. Helmi and P. T. D. Zeeuw. Mapping the substructure in the galactic halo with the next generation of astrometric satellites. *Mon. Not. R. Astron. Soc.*, 319(astro-ph/0007166):657, 2000.

[19] A. Inselberg. *Parallel Coordinates : VISUAL Multidimensional Geometry and its Applicationss*. Springer, New York, 2009.

[20] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3(1):1–58, 2009.

[21] E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating clustering in subspace projections of high dimensional data. *PVLDB*, 2(1):1270–1281, 2009.

[22] V. Petrosian. Surface brightness and evolution of galaxies. *Astrophysical Journal Letters*, 209:L1–L5, Oct. 1976.

[23] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali. A Monte Carlo algorithm for fast projective clustering. In *ACM SIGMOD*, pages 418–427, 2002.

[24] P. Salembier, A. Oliveras, and L. Garrido. Anti-extensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7:555–570, 1998.

[25] P. Salembier and J. Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing*, 4:1153–1160, 1995.

[26] P. Salembier and M. H. F. Wilkinson. Connected operators: A review of region-based morphological image processing techniques. *IEEE Signal Processing Magazine*, 26(6):136–157, 2009.

[27] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, New York, 1982.

[28] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[29] M. A. Westenberg, J. B. T. M. Roerdink, and M. H. F. Wilkinson. Volumetric attribute filtering and interactive visualization using the max-tree representation. *IEEE Transactions on Image Processing*, 16(12):2943–2952, 2007.

[30] M. H. F. Wilkinson and B. C. Meijer. DATAPLOT: A graphical display package for bacterial morphometry and fluorimetry data. *Comp. Meth. Prog. Biomedicine*, 47:35–49, 1995.

[31] M. Zeilik and S. Gregory. *Introductory Astronomy and Astrophysics*. Brooks/Cole astronomy list, Thompson Learning, 4th edition, 1998.