Fast Multiscale Clustering and Manifold Identification

Dan Kushnir *, Meirav Galun, Achi Brandt

The Weizmann Institute of Science, Department of Computer Science and Applied Mathematics,
Rehovot 76100, Israel.

Abstract

We present a novel multiscale clustering algorithm inspired by algebraic multigrid techniques. Our method begins with assembling data points according to local similarities. It uses an aggregation process to obtain reliable scale-dependent global properties, which arise from the local similarities. As the aggregation process proceeds, these global properties affect the formation of coherent clusters. The global features that can be utilized are for example density, shape, intrinsic dimensionality and orientation. The last three features are a part of the manifold identification process which is performed in parallel to the clustering process. The algorithm detects clusters that are distinguished by their multiscale nature, separates between clusters with different densities, and identifies and resolves intersections between clusters. The algorithm is tested on synthetic and real datasets, its running time complexity is linear in the size of the dataset.

Key words: algebraic multigrid (AMG), aggregation, graph partitioning, similarity-based clustering, manifold, data analysis, astrophysical models

1 Introduction

Clustering algorithms are useful in many fields, from image analysis through astronomy to biology. Generally, clustering is applied to a dataset, which is a collection of N d-dimensional vectors (data points) representing d measured features per sample. Given a dataset, clustering algorithms seek a partition of the data to coherent groups, in a sense that data points in the same group share similar properties. Many approaches try to solve the clustering problem by optimizing a global cost function, expressed in terms of the local similarities between data points.

Typical datasets contain clusters that differ from each other in density, and may also contain elongated clusters that may intersect. Moreover, in many cases clusters of interest include

* Corresponding author: Dan Kushnir, Tel 972-8-9343251, Fax 972-8-934-4122.
Email address: dan.kushnir@weizmann.ac.il (Dan Kushnir).

points that represent noisy samples from some underlying manifold structures. Also, many datasets are multiscale in nature, containing a nested structure of small clusters within larger clusters. In the scope of this work, we attempt to separate between clusters with different densities, to identify intersections between clusters, to support the creation of clusters that preserve smooth continuation, and in particular to distinguish between the different clusters that cross an intersection. To realize these objectives and to discriminate between the different clusters at all scales in the presence of noise, scale-dependent global properties should affect the formation of coherent clusters. The main global features that are utilized in our present study are density, shape, intrinsic dimensionality and orientation. The last three features are part of the manifold identification process, which is performed in parallel to the clustering process.

The importance of integrating these global features into the clustering process is exemplified in Fig. 1. It should be emphasized that a variety of additional global features, called also multiscale similarity features or aggregative properties, can be integrated into the process. See for example [14,27], where in the problem of image segmentation there enter aggregative properties such as average color, color variations at all intermediate scales, boundary match, shape properties of salient sub-aggregates, etc.

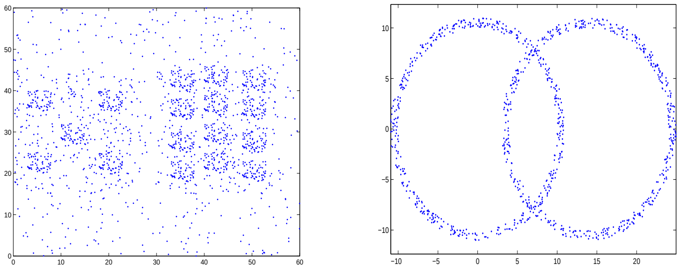

Fig. 1. **Nested clusters** (left): the different distributions of the dense patches at large scale give rise to separation of two different clusters. **Intersection of elongated clusters** (right): separation of intersecting shapes is possible by tracking their orientation at large scales.

In this paper, we present a novel clustering algorithm inspired by algebraic multigrid techniques (AMG) [5]. At the basis of our methodology is the normalized-cut cost minimization [30], in the sense that salient clusters in the dataset can be characterized by low normalized-cut costs. The minimization problem can be formulated as a generalized eigenvector problem.

Many other approaches that attempt to solve clustering problems (including spectral clustering methods e.g. [1]) formulate the problem as a generalized eigenvector problem, and usually solve them by using an eigensolver with quadratic or cubic complexity. An efficient way in most practical cases to compute eigenvectors at just a linear complexity is offered by an AMG eigensolver [6]. It is important to realize, however, that the AMG solver contains itself a hierarchical aggregation procedure which already yields a (hierarchical) clustering of the dataset, and that it is much better to directly use this procedure for clustering than actually computing the eigenvectors, or using other clustering procedures. This is because:
(1) If desired, this procedure can yield the same clustering as computed from the eigenvectors, at a smaller cost.

(2) This procedure will actually yield a **hierarchical** clustering, breaking the clusters into sub-clusters, sub-sub-clusters, etc.
(3) The produced clusters can be **fuzzy**, with some data points remaining undecided, belonging with different probability weights to different clusters.
(4) Most important, the hierarchical aggregation procedure can be modified to account for similarities in **global properties** of aggregates that emerge at various intermediate levels (scales). To our best knowledge such use of multiscale similarity features cannot be considered by any uniscale procedure, or for that matter, by any formulation of the problem as a functional minimization problem.
(5) Top-down procedures can easily be iteratively incorporated at all levels, to affect finer-level aggregation criteria by properties found important at coarser levels.

Our AMG-like approach discovers the desired aggregation of the dataset by following the similarities between the data points at different resolutions, using (similarly to [27]) a bottom-up weighted aggregation coarsening procedure that preserves the low normalized-cut costs. Moreover, to achieve coherent clusters at all scales, our approach allows combining multiscale similarity features, based on properties of aggregates that emerge at intermediate levels. The combined approach of bottom-up weighted aggregation and multiscale similarity features constructs a hierarchical pyramid of aggregates such that a salient cluster is guaranteed to emerge at a certain appropriate level with low normalized-cut cost. The cost of the algorithm is linear in the dataset size, and is independent of the number of clusters.

Clustering and manifold identification are known to be related. In manifold learning one is interested in finding the intrinsic dimensionality and low dimensional structure of the data. In this work, the clustering and manifold identification processes influence each other, so that the cluster partition is used to approximate the manifolds, and the manifold structure is used to improve the cluster partition. The identification of manifolds created by aggregates is dealt within the bottom-up process by using a scale-dependent local principal component analysis (PCA). An aggregate manifold is represented as a composition of spatially ordered sub-manifolds, each of which is approximately convex and well approximated by a set of principal axes. The aggregate manifold is identified even in the cases in which the manifold is non-convex and noisy.

In addition to the bottom-up aggregation process, a top-down process is applied in the present work to resolve intersections between clusters and to separate dense clusters from background noise. Relying on the AMG strength, the algorithm can be applied to datasets of any dimensionality, although the junction resolving, which relies on smooth continuation of orientations, is currently developed only for the cases of clusters with intrinsic dimensionality of 2D and 3D. The complexity of the algorithm is not dependent on the data dimensionality.

The paper is divided as follows. In section 2 we describe work related to clustering algorithms and manifold learning. In section 3 an overall description of the clustering algorithm is given. In section 4 we demonstrate the use of aggregative properties. In section 5 the algorithm complexity analysis is presented. In section 6 clustering results of real astrophysical data in 3D are demonstrated. Section 7 compares our 2D and 3D results with results obtained by other algorithms.

2 Related work

There are numerous approaches for data clustering and manifold learning. In this survey we mostly refer to the algorithms that are related to our approach. For an extensive overview see text books such as [9,21].

Spectral clustering methods for graph-based clustering and image segmentation [1,30] use the eigenvectors of the Laplacian matrix to embed the data in a lower subspace where they are expected to be well separated. Spectral clustering uses explicitly eigenvector solvers to find clusters whose graph cut is minimal. The Nyström approximation [12] is used to decompose the similarity matrix W efficiently by choosing a random sample of the data, so that the complexity of decomposing W to its eigenvectors is of $O(nN)$ where n is the size of the sample, and N is the size of the dataset. Path-based methods [10,11] discover elongated structures and overcome noise. Their complexity is at least $O(N^3)$. In the super-paramagnetic clustering (SPC) [4] method, also known as the granular magnet method, the data points are modelled as a collection of magnets. The scale of the temperatures used in a Monte-Carlo simulation of this collection determines the resolution at which the magnets align to form clusters, creating hierarchical clustering similar to ours. A related work [15] approaches the clustering problem as a minimal cut problem and produces a stochastic set of cuts by hard contractions of the original graph. The complexity of [15] is $O(Nlog^2(N))$. In tensor-voting [17] additional properties of location and orientation of data points in 2D and 3D are used to cluster data points and characterize their manifold. The method also detects cases of junctions and copes well with noise. Moreover, a criterion to measure smooth continuation between oriented structures, which is based on proximity and curvature, has some similarities with our criterion of completion probability (section 4.2). Tensor voting divides the data into voxels, each voxel aggregates some data points and geometric features but only on the scale induced by the partition into voxels. The algorithm complexity is $O(n^3k)$ where n is the number of voxels at the side length of the dataset volume, and k is the number of the additional input properties.

Some of the early manifold learning methods are the principal component analysis (PCA) [22] and multi-dimensional scaling (MDS) [7]. In extensions of PCA such as the principal curve method (and the principal surface method) [19,23,31] one estimates a manifold by computing a smooth curve that passes through the ”middle” of a *d*-dimensional data cloud. In projection pursuit [13] different optimization strategies are sought to find a basis for local data projections that optimize certain criteria. Our use of scale-dependent local PCA in different scales reveals the shape of the clusters with respect to their scatter. The local-linear embedding (LLE) algorithm [26] learns the manifold structure by finding a global coordinate system on the manifold. LLE attempts to compute a low dimensional embedding such that nearby points in the high dimensional space remain nearby and similarly co-located in the low dimensional space. A similar approach is the Laplacian eigenmaps (LEM) method

[2], where the graph Laplacian matrix is used for dimensionality reduction that preserves local proximity. In Isomap [33] the embedding is optimized with the constraint of preserving geodesic distances. The complexity of LLE, LEM and Isomap is at least $O(N^2)$.

3 The clustering algorithm

The clustering problem can first be formulated as seeking for a minimal normalized-cut in a weighted graph. Given a dataset with N data points and a distance matrix D of the dissimilarities between the data points, a weighted graph $G = (V, W)$ is constructed as follows. Each data point i is represented by a graph node $i \in V$ where $V = \{1, 2, ..., N\}$. For every two nodes i and j the following similarity weight is assigned

$$w_{ij} = \exp(-c_{dist} \cdot d_{ij}), \tag{1}$$

where c_{dist} is a pre-defined parameter that is determined with experience, and d_{ij} is usually the Euclidean distance between data points i and j; w_{ii} is set to 0. This constructs the similarity matrix $W = \{w_{ij}\}$. To evaluate clusters we define a saliency measure as follows. Every cluster $S \subseteq V$ is associated with a state vector $u = (u_1, ..., u_N)$ representing the assignments of data points to a cluster S

$$u_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S. \end{cases} \tag{2}$$

The **saliency** associated with S is defined by the **normalized-cut** cost

$$\Gamma(S) \overset{def}{=} \frac{\sum_{i>j} w_{ij} \cdot (u_i - u_j)^2}{\sum_{i>j} w_{ij} \cdot u_i \cdot u_j}, \tag{3}$$

which sums the weights along the boundaries of S divided by the internal weights. Clusters with small values of $\Gamma(S)$ are considered salient. In matrix notation Γ can be written as

$$\Gamma(S) = \frac{u^T L u}{\frac{1}{2} u^T W u}, \tag{4}$$

where L is the Laplacian matrix [30,14] whose elements are

$$l_{ij} = \begin{cases} \sum_{k\ (k \neq i)} w_{ik} & i = j \\ -w_{ij} & i \neq j. \end{cases} \tag{5}$$

If we allow arbitrary real assignments to u, then the minimum of Γ can be obtained by the minimal generalized eigenvector u of $Lu = \lambda W u$ $(\lambda > 0)$. Our objective is to find those partitions characterized by a small value of Γ.

Although an eigensolver, in particular an AMG eigensolver [6], can be applied to explicitly solve the generalized eigenvalue problem, we solve the clustering problem by an AMG-like approach (see [5]) without explicit computation of the eigenvectors. Our AMG-like procedure seeks salient clusters by following the similarity of the data points at different resolutions, from fine scales to coarser ones. Moreover, to further separate clusters at all scales, our AMG-like approach calculates and incorporates multiscale similarity features (e.g. density, shape, intrinsic dimensionality and orientation), which are called **aggregative properties**. As a result a hierarchical pyramid of graphs is constructed. Each node, at a certain scale, represents an **aggregate**, which is a weighted collection of the original data points. Each **cluster** S, which is a **salient aggregate** (i.e., $\Gamma(S)$ is low) emerges as a single node at a certain scale.

3.1 Multiscale graph coarsening: weighted aggregation

Starting from the given graph $G^{[0]} = G$, we recursively coarsen the minimization problem, creating the sequence of graphs $G^{[1]}, ..., G^{[k]}$ of decreasing size. At each scale we seek for nodes with low Γ. The salient aggregates, or clusters, represented by low-Γ nodes, are considered as approximate solutions to the minimization problem. As in the general AMG setting, the choice of the coarse variables (”C-points”), the design of the fine-to-coarse aggregation (or coarse-to-fine interpolation), and the derivation of the coarse problem are determined automatically, as described below.

Although the AMG approach can handle the full graph $G^{[0]} = G$ as defined above, the complexity of the algorithm is lowered by applying a dilution procedure which sets to 0 every w_{ij} that is relatively small. We first apply to $G^{[0]}$ the k-nearest neighbors algorithm (KNN) [3] (typically $10 \leq k \leq 50$). In addition to KNN we apply the following edge dilution procedure [16]: for each pair of neighboring nodes i and j we set w_{ij} to 0 in case $w_{ij}/\ \sum_{k:\langle i,k\rangle} w_{ik} < \gamma$ and $w_{ij}/\ \sum_{k:\langle j,k\rangle} w_{jk} < \gamma$ (or $w_{ij}/\ \max_{k:\langle i,k\rangle}\{w_{ik}\} < \gamma$ and $w_{ij}/\ \max_{k:\langle j,k\rangle}\{w_{jk}\} < \gamma$), in our experiments γ is set to 0.1. The edge dilution procedure can be applied at each pyramid level.

The construction of a coarse graph from a given one is divided into three stages:
(1) A subset of the fine nodes is chosen to serve as the **seeds** of the aggregates (the later being the nodes of the coarse graph).
(2) The rules for interpolation are determined, thereby establishing the fraction of each non-seed node belonging to each aggregate.
(3) The weight of the edges between the coarse nodes is calculated.
Coarse nodes. The construction of the set of seeds C (”C-points”) and its complement denoted by F, is guided by the principle that each F-node should be ”strongly coupled” to C. To achieve this objective we start with an empty set C, hence $F = V$, and sequentially (according to decreasing aggregate size defined in section 3.2) transfer nodes from F to C until all the remaining $i \in F$ satisfy $\sum_{j \in C} w_{ij} \geq \alpha \sum_{j \in V} w_{ij}$, where α is a parameter, in most experiments $\alpha = 0.2$.
The coarse problem. Each node in the chosen set C becomes the seed of an aggregate that will constitute one coarse scale node. We define for each node $i \in F$ a coarse neighborhood $N_i = \{j \in C, w_{ij} > 0\}$. Let $I(j)$ be the index in the coarse graph of the node that rep-

```
• Procedure CoarsenGraph(s,$\alpha$):
  · Initialize the set of seeds ($C$) and its complement ($F$): $C = \emptyset,\ F = V^{[s-1]}$.
  · for all nodes $i \in F$:
        if ($\sum_{j \in C} w_{ij}^{[s-1]} < \alpha \sum_{k \in V^{[s-1]}} w_{ik}^{[s-1]}$):
            $C \leftarrow \{C \cup \{i\}\};\quad F \leftarrow \{F \setminus \{i\}\};$
  · Calculate $P^{[s-1][s]}$, the interpolation weights (6).
  · Calculate $W^{[s]}$, the coupling weights (7).
```

Fig. 2. **Graph coarsening pseudo-code procedure.**

resents the aggregate around a seed whose index at the fine scale is j. The classical AMG interpolation matrix P (of size $N \times n$, where $n = |C|$) is defined by

$$P_{iI(j)} = \begin{cases} w_{ij} / \ \sum_{k \in N_i} w_{ik} & \text{for } i \in F, j \in N_i \\ 1 & \text{for } \ i \in C, j = i \\ 0 & \text{otherwise}. \end{cases} \tag{6}$$

It satisfies $u \approx PU$, where $U = (U_1, U_2, ..., U_n)$ is the coarse level state vector. P_{iI} represents the likelihood of i to belong to the $I-$th aggregate. Following the weighted aggregation scheme [27], the edge connecting two coarse aggregates p and q is assigned with the weight

$$w_{pq}^{coarse} = \sum_{k \neq l} P_{kp} w_{kl} P_{lq}. \tag{7}$$

w_{pq}^{coarse} is also called the **coupling weight** between aggregates p and q. Intuitively, the coupling weight between a pair of coarse aggregates (left hand side of (7)) is the weighted sum of the coupling weights between their sub-aggregates (right hand side of (7)). Using the interpolation matrix P, the saliency measure (4) can be written as

$$\Gamma = \frac{u^T L u}{\frac{1}{2} u^T W u} \approx \frac{U^T P^T L P U}{\frac{1}{2} U^T P^T W P U}. \tag{8}$$

The right hand side of (8) determines a coarser graph with n nodes whose similarity matrix is $W^{coarse} = P^T W P$. Exploiting the sparseness of P, the elements of $P^T W P$ are inexpensive to calculate. $L^{coarse} = P^T L P$ is approximated by a relation to W as in (5) [14].

This coarsening procedure is performed recursively. We denote a coarse scale by s, and its predecessor finer scale by $(s-1)$. The scale index is attached to the graph notation, i.e. a graph at scale s is denoted by $G^{[s]} = (V^{[s]}, W^{[s]})$, the appropriate interpolation matrix between scale s and $(s-1)$ is denoted by $P^{[s-1][s]}$ or $P^{[s-1]}$, and $|V^{[s]}|$ is denoted by $N^{[s]}$. A summary of the coarsening procedure is given in Fig. 2.

3.2 Aggregative properties

Consider a specific clustering problem in which small clusters are nested within larger clusters, as exemplified in Fig. 4. Multiscale use of the densities may reveal the nested structure and eventually cluster the data appropriately. In Fig. 4 all the dense patches have similar average density, yet in region **A** the patches are distributed sparsely, and in region **B** they are more tightly packed. Hence, at a small scale, the dense patches should be grouped together, yet on a larger scale regions **A** and **B** should be partitioned into different clusters as indeed achieved by our algorithm and shown in Fig 4.

In the case of astrophysical flow simulations, where at a given moment each star has a defined location as well as velocity, an important example of aggregative property is the average velocity of an aggregate. While the velocities of individual stars may be quite chaotic, their averages are significant and intermediate-level aggregates with similar averages (and other matching moments) should be grouped together to give a coherent view of the flow.

Since our weighted aggregation framework allows to aggregate a variety of multiscale properties, we call these properties aggregative properties. In this framework, for each aggregate i emerging at a certain scale s, we calculate a set of aggregative properties. An aggregative property can be expressed as a weighted average over the aggregate i of a property that has first appeared at a scale r $(r \leq s)$. The scale s is termed the **aggregate scale** and the scale r is called the **property scale**. At each scale s the similarity matrix $W^{[s]}$, inherited from finer aggregate scales (7), is modified by the similarities arising from the set of aggregative properties obtained from multiple property scales. In the scope of this work aggregative properties of density, shape, dimensionality and orientation are computed and incorporated. The aggregative properties are used to obtain partition into clusters that differ in density, to separate background noise from clusters, and to resolve intersecting clusters. Moreover, they are utilized to identify the manifold that approximately span each aggregate. As a straightforward aggregative property the reader may consider the center of mass of an aggregate. For an original data point the center of mass is simply its spatial coordinates (in this case $r = 0$, $s = 0$). For an aggregate i at scale 1 the center of mass is the weighted average of the spatial coordinates induced by the data points associated with aggregate i, the weights being the interpolation weights (in this case $r = 0$, $s = 1$). Similarly, the center of mass of an aggregate at scale s is a weighted average of the center of mass of its sub-aggregates at scale $(s-1)$ associated with it. This is exactly the center of mass as if explicitly calculated from the cloud of data points that assembles this aggregate.

The following formulas are applied to compute the aggregative properties. Let a property Q appear at the property scale r, where its set of values is $Q^{[r][r]} = q^{[r]} = (q_1^{[r]}, ..., q_{N^{[r]}}^{[r]})$. Then the average of Q over aggregate k at scale s is given by $\bar{Q}_k^{[r][s]} = \sum_j p_{jk}^{[r][s]} q_j^{[r]} \big/ \sum_j p_{jk}^{[r][s]}$, where $p_{jk}^{[r][s]}$ is the (j,k) element in the product matrix $P^{[r][s]} = P^{[r]} \cdots P^{[s-1]}$, which is the fraction of aggregate j at scale r in aggregate k at scale s. A fast computation of an aggregative property can be achieved by utilizing the following recursive relation

$$Q^{[r][s]} \overset{def}{=} Q^{[r][s-1]}P^{[s-1]}, \;\; M^{[r][s]} \overset{def}{=} M^{[r][s-1]}P^{[s-1]}, \;\; (s > r) \tag{9}$$

where $M^{[r][r]} \overset{def}{=} \vec{1} = (1, \dots, 1)$ at length $N^{[r]}$. Note that $M_k^{[r][s]}$ is the number of sub-aggregates at scale r that compose the aggregate k at scale s. In particular, $M_k^{[0][s]}$, which is the number of data points which compose aggregate k at scale s, is called **aggregate size**. From these recursive relations one can then calculate the required weighted average:

$$\bar{Q}_k^{[r][s]} = \frac{Q_k^{[r][s]}}{M_k^{[r][s]}}. \tag{10}$$

In this way the aggregative properties at each level s are calculated from information already accumulated at the immediately preceding level ($s - 1$).

The geometrical volume. Let $x_i = (x_i^{(1)}, \dots, x_i^{(d)})$ be the coordinates of a data point i. The center of mass of aggregate k at scale s is denoted by $\bar{x}_k = (\bar{x}_k^{(1)}, \dots, \bar{x}_k^{(d)})$ and computed by (10), where $r = 0$ and $Q^{[0][0]} = (x_1, ..., x_N)$. The **weighted covariance** is the $d \times d$ matrix $\Sigma_k = \overline{(x - \bar{x}_k)^T(x - \bar{x}_k)}$, i.e., $(\Sigma_k)_{\mu\upsilon} = \overline{(x^{(\mu)}x^{(\upsilon)})_k} - \bar{x}_k^{(\mu)}\bar{x}_k^{(\upsilon)}$, where $\overline{(x^{(\mu)}x^{(\upsilon)})_k}$ is $\bar{Q}_k^{[0][s]}$ calculated by (10) with $Q^{[0][0]} = (x_1^{(\mu)}x_1^{(\upsilon)}, ..., x_N^{(\mu)}x_N^{(\upsilon)}),\ (\mu, \upsilon = 1, ..., d)$. PCA is applied to find an eigenvector basis $\{\vec{v}_k^{(1)}, \dots, \vec{v}_k^{(d)}\}$ of Σ_k and its corresponding set of eigenvalues $\{\lambda_k^{(1)}, \dots, \lambda_k^{(d)}\}$. The eigenvalues are used to approximate the geometrical volume of a convex aggregate k at a scale r as follows:

$$V_k^{[r][r]} = \prod_{i=1}^{d} \sqrt{\lambda_k^{(i)}}. \tag{11}$$

The geometrical volume of a non-convex aggregate k at scale s is approximated by the accumulated geometrical volume of its sub-aggregates, i.e., the k-th element of $V^{[r][s]} = V^{[r][r]} \cdot P^{[r][s]}$. The notions of convex and non-convex are explained in the next section.

The density. The density h of an aggregate i at scale s is defined by the ratio between the number of data points that compose this aggregate and the accumulated geometrical volume: $h_i^{[r][s]} = M_i^{[0][s]} / \, V_i^{[r][s]}$, where r is the scale at which PCA is applied (typically $r = 3$).
The typical distance of a data point i is the average Euclidean distance from its neighbors: $b_i = \frac{\sum_{j:\langle i,j\rangle} \|\bar{x}_i - \bar{x}_j\|}{n_i}$, where n_i is the number of neighbors. The typical distance $\bar{b}_i^{[0][s]}$ of an aggregate i at scale s is computed by using (10), for $Q^{[0][0]} = (b_1, b_2, ..., b_N)$. The typical distance is inversely related to the density.

3.3 Manifold identification

The aggregative properties are utilized to identify the manifold that span an aggregate and to reveal the intrinsic dimensionality of the aggregate. The manifold identification significantly

```
• Procedure IdentifyManifold(s):
  · for all nodes $k \in V^{[s]}$:
        compute the convexity measure.
        if $k$ is a convex aggregate
            calculate the fractional variance (12).
            define its manifold type by using Definitions 1,2.
        else
            if all sub-aggregates have the same manifold type $l$
                define the manifold type of $k$ to be $l$.
            else
                the manifold type is heterogenous.
                aggregate $k$ is denoted as a junction.
```

Fig. 3. **Manifold identification pseudo-code procedure.**

affect the formation of clusters with similar intrinsic dimensionality.

From a certain scale s (typically $s = 3$) we examine for each aggregate k a **convexity measure**: $V_k^{[r][s]} \big/ V_k^{[s][s]}$. If the ratio is less then c (in our experiments $c = \frac{1}{2}$) then the aggregate is considered as **non-convex**, otherwise it is **convex**. A type of a manifold is defined directly for a convex aggregate and indirectly (recursively) for a non-convex one. To characterize the manifold type for a convex aggregate k, the definitions below are used.

$$\mathit{FVAR}_k(i) \overset{def}{=} \frac{\lambda_k^{(i)}}{\sum_{j=1}^{d} \lambda_k^{(j)}} \quad i = 1, ..., d \tag{12}$$

denotes the fraction of variance obtained in the direction of $\vec{v}_k^{(i)}$, relatively to the total variance attained in all principal directions.

Definition 1 *A convex aggregate k is defined as* ***wide*** *in direction $\vec{v}_k^{(i)}$ if $FVAR_k(i) > \frac{\delta}{d}$, for a given δ (typically $\delta = 0.5$).*

Definition 2 *A convex aggregate is spanned by an* ***lD-manifold*** *if it is wide in l directions.*

In case the examined aggregate is non-convex, the manifold type is determined by its sub-aggregates (the finer level aggregates which form it). The sub-aggregates of aggregate k are scanned and if all of them have the same dimensionality l then the manifold type of aggregate k is defined to be l. Otherwise, the manifold type of the aggregate is defined to be **heterogenous** and identified as a **junction**. A manifold identification pseudo-code is given in Fig. 3. Manifold identification results are demonstrated in Fig. 5 for three different structures.

For scales at which aggregates become non-convex we define the notion of tips. Tips are those convex sub-aggregates that form endpoints of the manifold that spans the aggregate (see Fig. 6). Tips are currently defined only for 1D-manifolds.

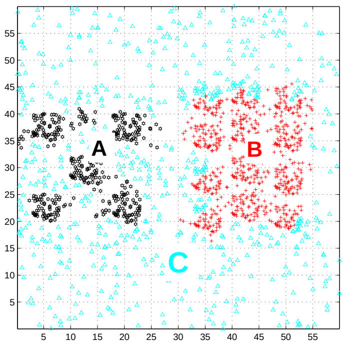

Fig. 4. **Nested clusters.** Each cluster discovered by our algorithm has a different color.

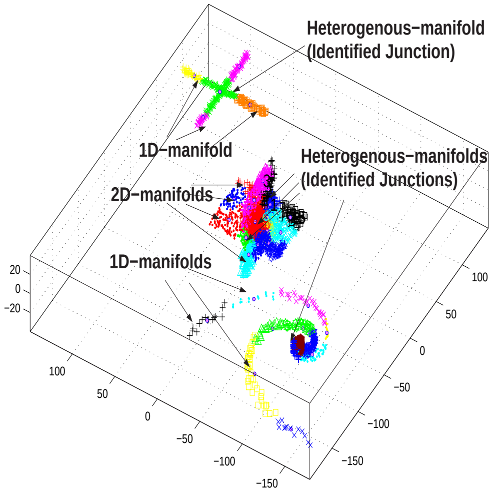

Fig. 5. **Manifold types.** Aggregates that emerged at a certain scale are displayed in different colors. The attached labels explain the manifold type that spans each of the aggregates.

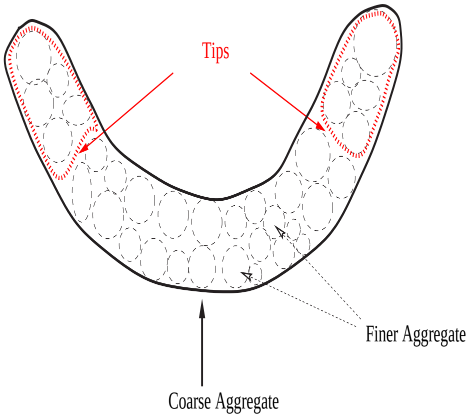

Fig. 6. **Tips of an aggregate**. The tips of a coarse aggregate are the intermediate scale sub-aggregates that resides at the ends of the aggregate manifold (red).

3.4 Algorithm outline

Aggregative properties of density and dimensionality are used to affect the aggregation so that fine aggregates that have similar dimensionality and density will merge to an aggregate on a coarser scale. Each aggregative property, obtained at a certain scale s, is formulated into a similarity measure between aggregates, and used to modify the coupling weights (7) between the aggregates at scale s. Those similarity measures are usually an expression depending on the difference between two aggregative properties (an absolute value of the difference or the square of the difference). The utilization of the similarity measures and the coupling modification formulas are elaborated in section 4. In addition to the bottom-up process, a top-down processes is used to split and merge aggregates of fine scales to correct inaccurate clustering which occurred during the bottom-up process. A more elaborated description of the top-down process is given in section 4.4.

The clustering algorithm is summarized by an outline in Fig 7, with the following parameters. The top-down procedure is performed at scale s_t (typically $6 \leq s_t \leq 9$) down to a finer scale (typically $r = 2$). The manifold identification is applied from scale r_{mn} and on (typically $r_{mn} = 3$). Aggregative properties reflect their similarity by modifying the coupling weights from scale s_c and on (typically $s_c = 3$).

```
• FMSC($\alpha$, $r$, $r_{mn}$, $s_c$,$s_t$)
(1)  Given a data set calculate $W^{[0]}$ - the similarity matrix (1).
(2)  Apply a dilution procedure.
(3)  $s = 1$.
(4)  while($N^{[s]} \neq N^{[s-1]}$)
         · CoarsenGraph(s,$\alpha$)
         · compute aggregative properties (sec. 3.2).
         · if ($s \geq r_{mn}$): IdentifyManifold($s$).
         · if ($s \geq s_c$): ModifyCoupling($s$).
         · if ($s = s_t$): TopDown($r$); $s \leftarrow r$.
         · $s = s + 1$.
```

Fig. 7. **The clustering algorithm outline.**

4 Determining coherent clusters

In this work we have focused on several specific objectives: to discriminate between clusters with different densities, to remove background noise that may incorrectly merge with clusters, to isolate junctions (intersections) between clusters, and to determine the different clusters that cross the junction in terms of smooth continuation (i.e., the manifold which is formed by each of the intersecting clusters has low curvature). So far we have described the aggregative properties that are accumulated through the bottom-up weighted aggregation. The way that we combine the aggregative properties to achieve those objectives is explained in this section.

4.1 The Mahalanobis distance

Given two aggregates k and l at scale s, with respective centers of mass $\bar{x}_k, \bar{x}_l$ and respective covariance matrices Σ_k, Σ_l, the mutual Mahalanobis distance is computed as follows. The **Mahalanobis distance** between $\bar{x}_l$ and the cloud of points in aggregate k is given by $Mahal(l,k) = \sqrt{(\bar{x}_k - \bar{x}_l) \cdot (\Sigma_k)^{-1} \cdot (\bar{x}_k - \bar{x}_l)^T}$. Similarly $Mahal(k,l)$ is defined. The Mahalanobis distance can be considered as a weighted Euclidean distance between a point and a cloud of points, where the relevant axes are the principal directions of the cloud and the weights reflect the spread of the data points on the principal directions. The **mutual Mahalanobis distance** is $Mut\_Mahal(k,l) \overset{def}{=} Max\{Mahal(k,l), Mahal(l,k)\}$. Starting from a certain scale s (typically $s = 3$) we bias the aggregation to preserve smooth continuation by multiplying the coupling weights (7) between any two neighboring aggregates k and l by $\exp(-c_M \cdot Mut\_Mahal(k,l))$, where in our experiments c_M is set to values between 1 and 10. Note that the use of the mutual Mahalanobis distance is restricted to cases where both neighboring aggregates are considered convex.

4.2 Completion probability and manifolds

The biasing by the mutual Mahalanobis distance promotes continuation between two colinear clouds of data-points. To promote also smooth continuation upon constant curvature, i.e. co-circularity, we rely on the Elastica criterion. The elastica criterion is used extensively in perceptual grouping works (e.g. [28,29,34]). We exploit the elastica criterion to support smooth continuation of aggregates and in particular to discriminate between clusters that cross a junction.

For two neighboring aggregates, the elastica criterion provides an estimation, called completion probability, that the two aggregates form a smooth continuation. The completion probability described for the 2D case in images [28] is generalized for 2D and 3D cases of scattered points.

The completion probability function in the 2D case estimates the probability that two neighboring aggregates k and l, which are $1D$-manifolds in R^2, form a smooth $1D$-manifold (Fig. 8). The completion probability is based on an energy function which is composed of two components. The first component is the ratio of the distance r' between the tips of the two aggregates and a radius of curvature ρ, defined by $\rho = (\frac{L_k^{(1)}+L_l^{(1)}}{2})^2 / (8 \cdot \frac{L_k^{(2)}+L_l^{(2)}}{2})$, where $L_i^{(1)}$ and $L_i^{(2)}$ are the length and width of aggregate i respectively ($L_i^{(1)} = \sqrt{\lambda_i^{(1)}}$, $L_i^{(2)} = \sqrt{\lambda_i^{(2)}}$), for $i = k, l$. The ratio r'/ρ is denoted by E_{dist}. The second component, denoted as E_{ang}^{ϕ}, is a function of the angles ϕ_k and ϕ_l, where $\phi_i \in (-\frac{\pi}{2}, \frac{\pi}{2})$ (for $i = k, l$), is the pitch angle of the 1^{st} principal direction of aggregate i with the line connecting the two centers of mass of the two aggregates. The square difference between the two angles $(\phi_k - \phi_l)^2$ reflects deviation from co-circularity, whereas their combined magnitude $(\phi_k^2 + \phi_l^2)$

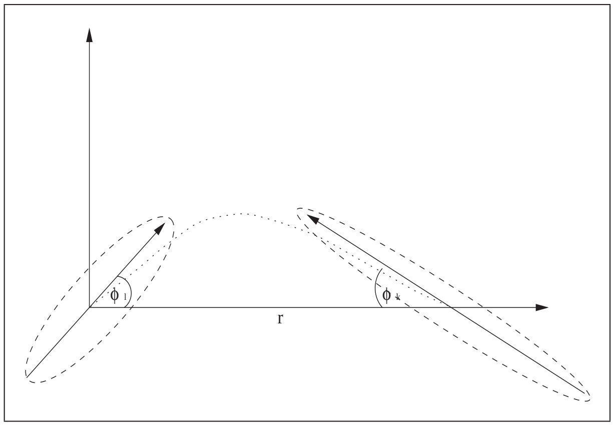

Fig. 8. **Completion between two aggregates in 2D.** Each aggregate (approximated by a dashed ellipse) creates an angle ϕ_i ($i = k, l$), between its 1^{st} principle axis and the line connecting the two centers of mass. The completion curve is drawn between the two centers of mass.

Expr.	ϕ_k	ϕ_l	$L_k^{(1)}$	$L_k^{(2)}$	$L_l^{(1)}$	$L_l^{(2)}$	r'	r	$G(k,l)$
A	0	0	2	0.08	2	0.08	0	4	1
B	0	0	2	0.08	2	0.08	0.5	4.5	0.96
C	0	0	2	0.08	2	0.08	3	7	0.57
D	30	30	2	0.08	2	0.08	3.5	7	0.24
E	30	-30	2	0.08	2	0.08	4.5	7	0.05
F	30	30	1	0.08	1	0.08	5.5	7	0.02
G	30	30	1	0.08	1	0.08	1.5	3	0.51
H	60	60	1	0.08	1	0.08	2	3	0.16

Table 1
A set of eight examples in 2D completion probability (see their display in Fig 9). In all examples $c_g = 0.8$, $c_d = 0.3$, $p_d = 1.5$, $p_g = 2$; k denotes the left aggregate and l denotes the right one.

reflects deviation from co-linearity. The energy of co-circularity and co-linearity is given by $E^{\phi}_{ang} = \frac{\rho}{r}\sqrt{\phi_k^2 + \phi_l^2 - \phi_k \cdot \phi_l}$, where r is the distance between the centers of mass of the aggregates. The completion probability between aggregates k and l at scale s in the 2D case is proportional to: $G^{[s]}(k,l) = \exp(-c_d \cdot (E_{dist}(k,l))^{p_d}) \cdot \exp(-c_g \cdot (E^{\phi}_{ang}(k,l))^{p_g}\ )$, where c_d, c_g, p_d and p_g, are pre-determined parameters (see Fig. 9 and Table 1). These parameters are quite robust for different datasets, yet, an automatic procedure to learn them may be developed in future work.

The completion curve. Given a pair of aggregates k and l that have high completion probability, a smooth completion curve that connects them can be constructed. The cubic spline approximates the elastica curve that minimizes the average curvature between a given pair of points p_1 and p_2, and their respective gradient values. In our context $p_1 = \bar{x}_k$ and $p_2 = \bar{x}_l$ are the centers of mass of k and l, and the gradient values are given by $\tan(\phi_k)$ and $\tan(\phi_l)$, respectively (see also [28])).

To encourage smooth continuation aggregation, the coupling weight between aggregates k and l is replaced by $max\{max_j\{w_{kj}\}, max_j\{w_{lj}\}\}$ when $G^{[s]}(k,l) > t$ for some predetermined threshold $0 \leq t \leq 1$. The completion probability is measured between aggregates that

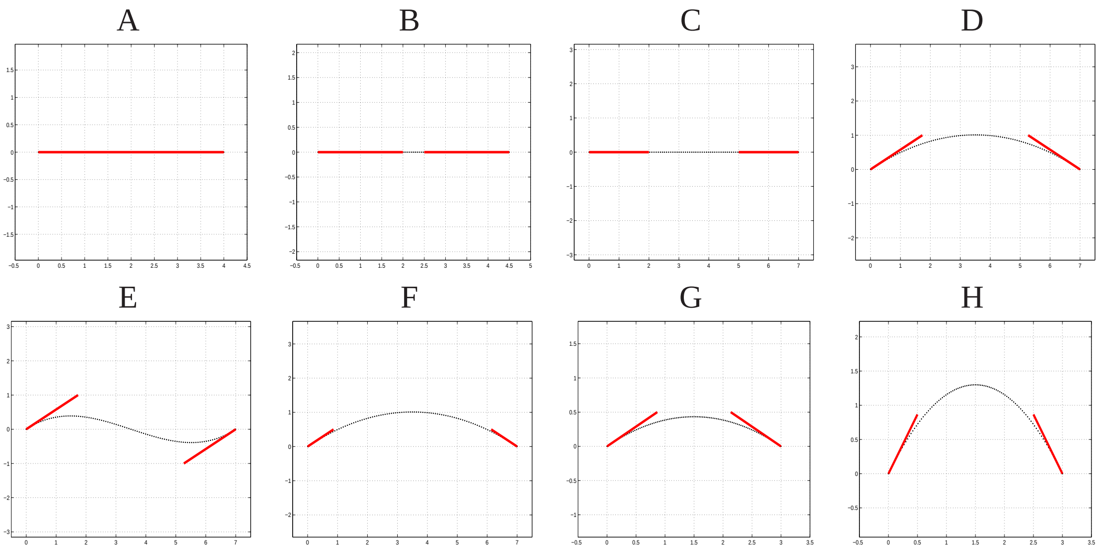

Fig. 9. **Completion of 1D-manifolds in 2D.** The thick lines are the 1st principal axis of each aggregate, the gray curve is the completion curve. The parameters for each example are given in Table 1.

are identified as convex 1*D*-manifolds. In case the aggregates are identified as 1*D*-manifolds but one of them is non-convex, the completion probability is measured between their convex tips (see section 3.3). The **3D case** of completion probability is explained in appendix A.

4.3 Density

In many tasks the density of data points is a meaningful criterion for separating clusters (e.g. [25,32]), and detecting sparse background noise. The aggregative properties of density and typical distance (subsection 3.2) are related measures. The density measure is utilized at the bottom-up process whereas the typical distance measure is used in the top-down refinement.

We have defined the density of aggregate i at scale s: $h_i^{[r][s]} = \frac{M_i^{[0][s]}}{V_i^{[r][s]}}$, which is the ratio between the number of elements that compose the aggregate and its accumulated geometrical volume. The variation of the density of an aggregate i is also of interest and defined by: $\sigma^2(h_i)^{[r][s]} = (\overline{(h^{[r][s-2]} - h_i^{[r][s]})^2})$, where $h^{[r][s-2]}$ denotes the density of the sub-aggregates of aggregate i at scale $s-2$. To support aggregation between aggregates with similar density the coupling weight between any two aggregates k and l at scale s is multiplied by

$$\exp\left(-c_{dens} \cdot |h_k^{[r][s]} - h_l^{[r][s]}| \diagup (\sigma(h_k)^{[r][s]} + \sigma(h_l)^{[r][s]})\right), \tag{13}$$

where c_{dens} is some non-negative constant; in our experiments c_{dens} is set around 10. Some examples are given in Fig. 10.

4.4 Top-down

In addition to the bottom-up process, a top-down process is performed to cure incorrect cluster partitions, according to global features which are detected only on coarse (top) scales. The **junction resolving** top-down process is used to determine the different manifolds that cross a junction, which has already been detected and isolated during the bottom-up process. The top-down process is based on the information obtained from the manifold identification: the junction and the orientation of manifolds. This high-level information obtained at a coarse scale s is used to modify the coupling weights at a finer scale r. Then the modification is followed by a second bottom-up process starting at scale r. An incorrect clustering result obtained from the initial bottom-up aggregation is demonstrated in Fig 11. The outline of the junction top-down resolution is as follows (see Fig 12 for illustration):

```
• For all neighboring aggregates of a suspected junction aggregate $i$ at scale $s$:
(1)  match a neighbor with high completion probability, and compute their completion curve
     (section 4.2).
(2)  strengthen coupling weights between $r$-scale sub-aggregates of aggregate $i$ which reside
     close to the completion curve.
• weaken coupling weights between junction $r$-scale sub-aggregates which do not reside on
  the same completion curve.
• detect the exact intersection domain: strengthen coupling weights between sub-aggregates
  that reside on more than one curve, weaken all their other coupling weights.
• Perform bottom-up aggregation starting at scale $r$.
```

A set of identified junction examples and their resolving is shown in Fig. 13.

The **density refinement** top-down process is used to detect and separate background noise which may mistakenly merge with clusters. The density refinement reflects the density information which is obtained at a coarse scale s and modifies the coupling weights at a finer scale r. Examples for inaccurate bottom-up clustering and cured top-down clustering are given in Fig. 14. The outline of the density refinement top-down procedure is given below:

```
• For aggregates $i$ at scale $s$:
(1)  compute typical distance $\bar{b}_i^{[0][s]}$ and standard deviation $\sigma(b_i)^{[0][s]}$ (section 4.3).
(2)  for all $r$-scale sub-aggregates $j$ of $i$:
     for all $k$ s.t. $w_{jk} > 0$:
         · if ($\bar{b}_j^{[0][r]} < \bar{b}_i^{[0][s]} - \sigma(b_i)^{[r][s]}$ or $b_j^{[0][r]} > \bar{b}_i^{[0][s]} + \sigma(b_i)^{[r][s]}$):
           weaken the coupling weight $w_{jk}^{[r]}$, according to the typical distance.
• Perform bottom-up aggregation from scale $r$.
```

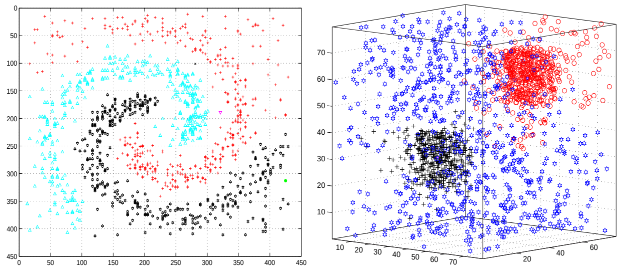

Fig. 10. **Applying the density criterion.** Dense regions are identified.

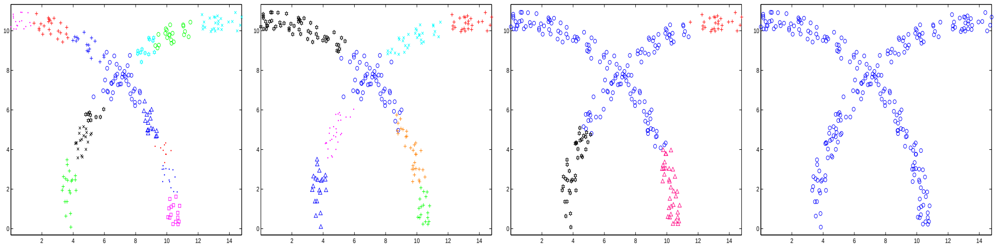

Fig. 11. **Bottom up aggregation.** Left to right: scales 4-7. A bottom-up aggregation without use of junction detection and resolution is demonstrated. The two intersecting clusters are not separated.

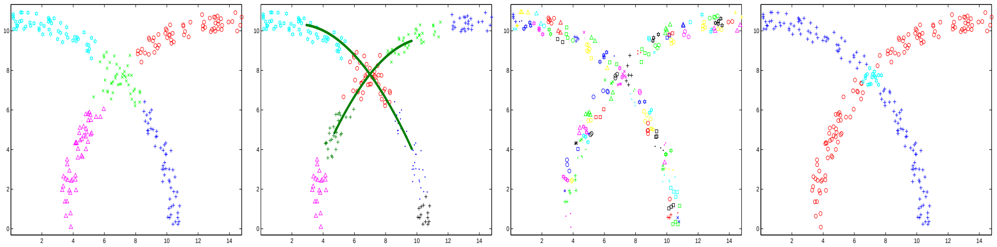

Fig. 12. **Top-down junction resolving.** Left to right: scale 8, coarsest scale of the first bottom-up process in which the junction aggregate has been identified; scale 5, sub-aggregates of the junction neighbors are matched to each other by completion probability, and completion curves are drawn; scale 2, fine aggregates are reclassified according to the completion curves; scale 6, coarsest scale of the second bottom-up process where the junction is resolved and the two intersecting clusters are separated.

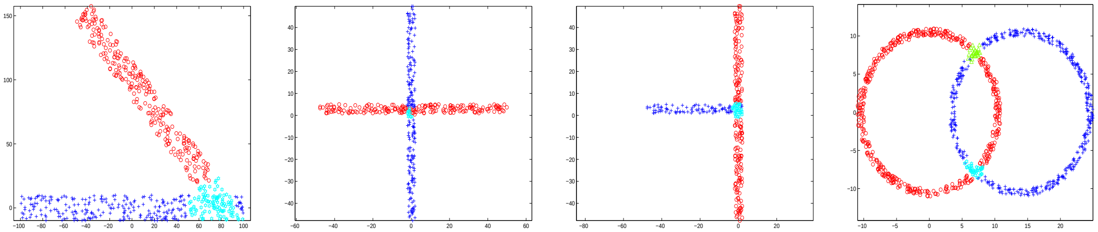

Fig. 13. **Four types of resolved junctions.** Junctions are resolved and the desired clustering is obtained by using the top-down process.

5 Complexity

The high complexity of a clustering algorithm can be a significant barrier when the datasets are large. Also, the preprocessing of calculating the local similarities between the data points can be expensive if done naively. In the scope of this paper, we do not intend to solve efficiently the preprocessing. We use the k-nearest neighbor (KNN) procedure to obtain a bounded number ($5 \leq k \leq 40$) of local similarities (neighbors) per a data point. In low dimension KNN has complexity of $O(N \log N)$, whereas in high dimension the complexity is $O(N^2)$, where N is the number of data points.

The complexity of our multiscale clustering algorithm is $O(N)$. At each scale s of the pyramid four steps are applied: the computation of coupling weights, the computation of the aggregative properties, the modification of $W^{[s]}$ according to the aggregative properties, and the choice of coarse representative for the next scale. Each of these steps has $O(N^{[s]})$ complexity. Therefore, $O(N^{[s]})$ operations are done at each scale s. A single graph coarsening step produces a coarse graph with about half the number of nodes of the finer graph. Thereby the complexity of one bottom up process is $O(N + N/2 + N/4 + ...) = O(N)$. The complexity of a top-down process starting at a top (coarse) scale s is influenced only by the number of operations performed on the fine scale r ($O(N^{[r]})$). Thus, the top-down process complexity is at most $O(N)$. Therefore, the total complexity of the algorithm is linear in the size of the dataset.

6 Clustering of astrophysical data

The fast growth of exploratory tools in astrophysics yielded massive data sets awaiting to be explored. Some of the underlying tasks in this field are the exploration of the different structures that galaxies and clusters of galaxies form both in real observations and in simulated models. This may serve as a key for determination of the underlying astrophysical model parameters that explain observations. The cold and dark matter (CDM) is assumed to be the major fraction of the universe mass. Cosmological simulations of the universe evolution are based on applying a dynamical model on CDM particles. As time evolves, CDM particles form peculiar structures such as filaments, sheets and spherical clusters, i.e. different manifold types. We use our algorithm in this context to demonstrate its capabilities of detecting structures in such datasets. We also use our algorithm to infer the fitting between an astrophysical model and real observations by comparing the distribution of different manifold types.

6.1 Cold and dark matter

The algorithm is applied on a sample of a 3D simulation which contains 20000 CDM particles (Fig. 15). The particles positions are the input data points for our algorithm. The advantages of using the aggregative properties and the manifold identification for discovering interesting structures are demonstrated in this example. In this dataset (Fig. 15) a dense plane, that is composed of dense cores, is sought to be separated from the surrounding sparse noise. Below the plane there is a sparser plane that is almost orthogonal to it. We have used KNN with $k = 40$. The use of the density feature and the Mahalanobis distance successfully detected

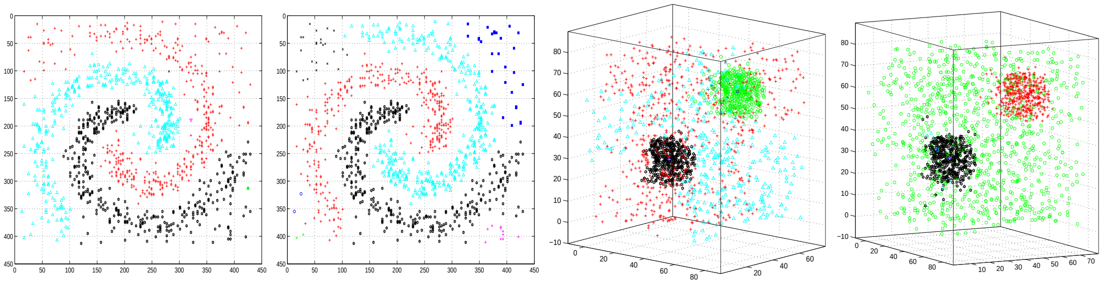

Fig. 14. **Inaccurate bottom-up clustering and cured top-down clustering.** Left to right: scale 9 after first bottom-up process obtained with misclassifications; scale 7, by applying top-down density refinement the background noise is separated from the clusters; scale 9, after first bottom-up process; scale 7, after applying top-down density refinement.

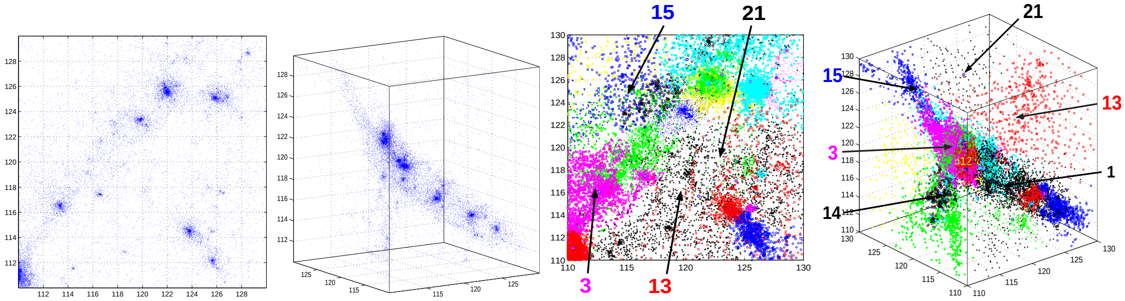

Fig. 15. **Clustering CDM simulation.** Left to right: a sample sub-cube of 20000 CDM particles, top and side views. Clustering of the data when utilizing the density and Mahalanobis features, top and side views.

the structures in the data. **Table** 2 demonstrates a comparison between our visual detection of structures and the manifold identification of our algorithm.

Cluster no.	1	3	13	14	15	21
Manifold Type	2D	2D	3D	2D	2D	3D
Visual detection	Dense Plane	Dense Plane	Noise	Lower Plane	Dense Plane	Noise

Table 2
Clustering a CDM simulation sample: identifying manifolds. The manifold type of selected aggregates shown in Fig. 15 is determined and compared to the visual detection of the shape patterns.

6.2 Comparison of a CDM model with a real observation

The use of our algorithm to compare an astrophysical model with a real observation is demonstrated. Specifically, we have checked a null hypothesis which claims that the multiscale distribution of manifolds of different dimensionality in an observation dataset is similar to the distribution in 22 model-based realization data. We have used as an observation the 2dF galaxy red-shift survey (2dF) and 22 ΛCDM model realizations (i.e. those are 22 datasets of a CDM model simulations). The model and observation have been already found similar with respect to density based criteria [20,24].

The probability of an aggregate to be an iD-manifold; $i = 0, 1, 2, 3$ is computed at each scale s (denoted by $\Psi_i^s(\Omega_j)$) for each model realization Ω_j, $j = 1, ..., 22$, and for the 2dF data set. Sparse clusters whose manifold can not be identified, due to small number of galaxies belonging to it, are also counted and considered as $0D$-manifolds. The average probability for a manifold of type i over all model realizations is computed (denoted by $E\{\Psi_i^s\}$), i.e. $E\{\Psi_i^s\} = \sum_{j=1}^{22} \Psi_i^s(\Omega_j)/\ 22$ is the average fraction, over all 22 model realizations, of the clusters found at scale s, whose manifold is an id-manifold. The standard deviation of this measure is also computed (denoted by σ_i^s). The standardized variable matrix 4×8 which measures the deviation of the 2dF manifolds distribution from the model manifolds distribution at 8 different scales of our algorithm is computed as $Z(i, s) = (E\{\Psi_i^s\} - \Psi_i^s(2dF))/\ \frac{\sigma_i^s}{\sqrt{22}}$. For all i and s $Z(i, s)$ values satisfy $|Z(i, s)| < 1.59$, which confirms with a 95% confidence interval that the observation is fitted by the model.

7 Comparison of algorithms

Several clustering algorithms and our fast multiscale clustering algorithm (FMSC) were applied to examples which are reported in relevant literature and to the CDM example. The algorithms compared are single-linkage [9] (SL), k-means [18], SPC [4] (also known as the granular magnet method), spectral clustering (SC) [1], connectivity-kernel clustering (CKC) [10], an algorithm that uses the k-means algorithm with the expectation maximization (EM) algorithm [8] (KEM), and our FMSC. In all examples the data points which belong to the same cluster are displayed in the same color. In Figs. 16-17 we demonstrate our detection of a junction, and the separation between the clusters that cross the junction. It may be required that connected components will be clustered as one cluster. Such a task can be achieved by our algorithm (as shown in Fig. 11) as well as by other algorithms. However, our intention is to demonstrate how FMSC uses the manifold identification and the orientation of aggregates to separate clusters even when ambiguity in cluster assignment exists, i.e a junction. In Figs 18-19 we compare the performance of different algorithms in separating dense clusters from noise. Of particular interest is the comparison demonstrated in Fig. 19, where we detected curved and elongated clusters and separated them from noise. In Fig. 20 we have found some of the underlying structures, yet some improvement in the use of aggregative properties is required in order to separate the whole dense plane in this sample. The k-means results shown in Fig. 20 have completely misclassified the dense clusters. Some of the other algorithms run out of memory resources when tested on the CDM example. Indeed, the comparisons manifest the need for aggregative properties in order to achieve desired clustering.

8 Conclusions

We presented a novel multiscale clustering algorithm, inspired by algebraic multigrid (AMG). Our AMG-like approach discovers the desired aggregation of the dataset by following the

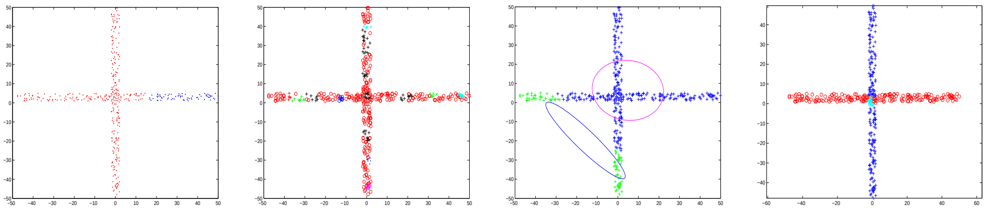

Fig. 16. **Clustering an ’X’ shape.** Left to right: SL, SPC, KEM - a bad execution, FMSC.

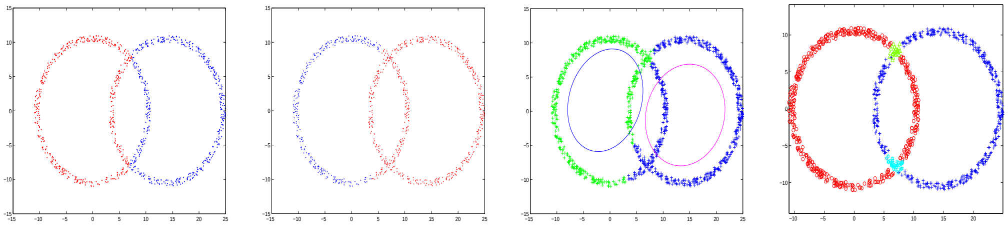

Fig. 17. **Clustering two intersecting circles.** Left to right: k-means, SC, KEM, FMSC.

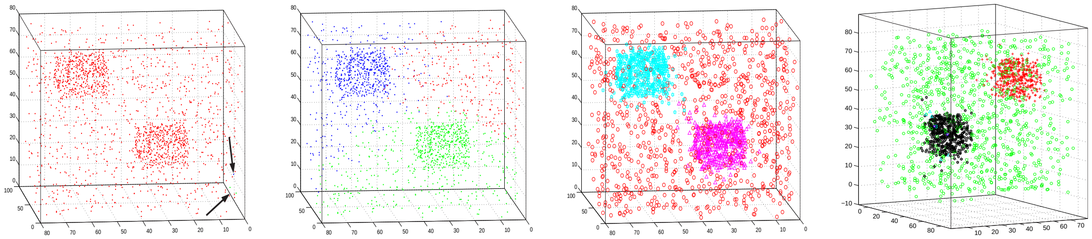

Fig. 18. **Clustering two dense clusters with noise.** Left to right: SL (two small clusters are indicated by arrows), k-means (k=3), SPC, FMSC.

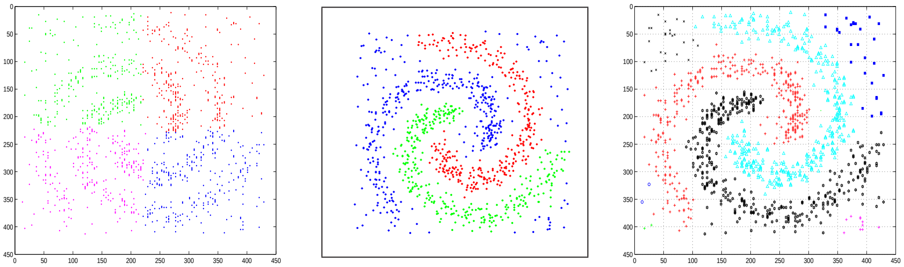

Fig. 19. **Clustering dense spirals cluttered with noise.** Left to right: k-means (k=4), CKC: (the figure is taken from http://www.inf.ethz.ch/personal/befische/nips03/), FMSC.

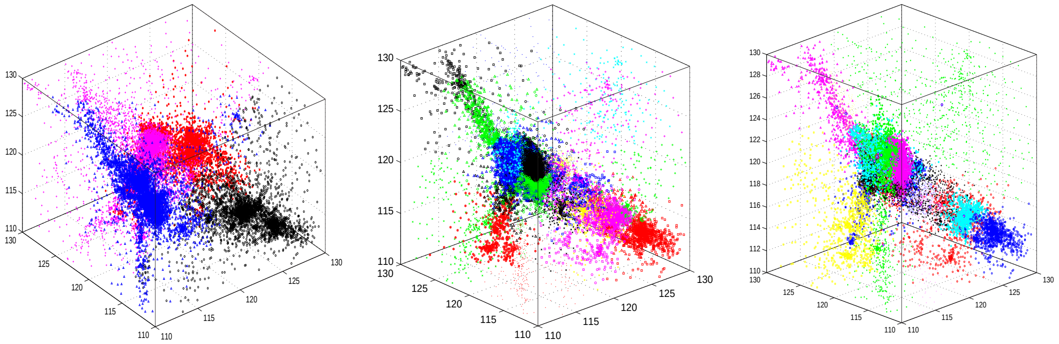

Fig. 20. **Clustering a CDM example.** Left to right: k-means k=4, k-means k=30, FMSC.

similarities between the data points at different resolutions, using a bottom-up weighted aggregation process. Moreover, to achieve coherent clusters at all scales, our approach uses multiscale similarity features and incorporates manifold identification processes. The algorithm detects clusters that are distinguished by their multiscale nature, separates between clusters with different densities and identifies and resolves intersections between clusters. The flexibility of our algorithm which allows to combine other statistics, i.e. additional multiscale similarity features, along with its low complexity, offer a powerful tool for exploring massive datasets.

9 Acknowledgments

This research has been supported by the Israel Science Foundation Grant no. 295/01, by the German-Israel Foundation for scientific research and development (GIF) Grant no. I-718-135.6/2001, and by the European Commission Project IST-2002-506766 Aim Shape. We thank Jean-Luc Starck from Service d'Astrophysique, CEA/Saclay, France, for giving us astrophysical datasets. We also thank Ronen Basri, Dorit Ron, Boaz Nadler, Beverly Lewin, and our anonymous reviewers for their insightful remarks.

APPENDIX A: Completion probability in 3D.

In the **3D case** of completion probability there are two situations: 1D-manifolds and 2*D*-manifolds in R^3 space. In the 1*D*-manifold case (Fig. 21) two angles are defined: the pitch angle ϕ between the $Z = 0$ plane and the 1^{st} principle axis, and the yaw angle ψ between the $Y = 0$ plane and the 1^{st} principle axis. The completion probability function reflects co-circularity and co-linearity for both angles: $G^{[s]}(k,l) = \exp(\ -c_d \cdot (E_{dist}(k,l))^{p_d} - c_g \cdot ((E^{\phi}_{ang}(k,l))^{p_g} + (E^{\psi}_{ang}(k,l))^{p_g})\ )$. In the 2D-manifold case (Fig. 21) the aggregates manifold structure is approximated by a plane. Given aggregates k and l, two angles are considered: the aggregate's roll angles difference (2δ), and their pitch angles (ϕ). The roll angles difference is measured as the angle formed between the directions of the intersection lines of the aggregate's manifolds with the $X = 0$ plane and $X = r$ plane respectively, (the lines are denoted by $I(k)$ and $I(l)$ where $I(l)$ is located at l center). We then fix the $Z = 0$ plane to intersect $X = 0$ where the bisector of the directions of $I(k)$ and $I(l)$ lays. The pitch angle of an aggregate k (denoted by $\phi(k)$) is measured as follows: the intersection $I(y,k)$ of $Y = 0$ with the planar manifold of k, is computed. $\phi(k)$ is the angle between $I(y,k)$ and the X-axis. Similarly, $I(y,l)$ and $\phi(l)$ are computed. The probability function G reflects co-circularity and co-linearity of the pitch angles, the difference in the roll angle 2δ, and the distance between the two aggregates: $G^{[s]}(k,l) = exp(-c_d \cdot (E_{dist}(k,l))^{p_d} - c_p \cdot (E^{\phi}_{ang}(k,l))^{p_g} - c_r \cdot |\delta|^{p_\delta})$, where c_d, c_p, and c_r are predefined parameters. ρ is computed by using ρ with $L_i^{[1]}$ equal to the length of $I(p+,i)$, and $L_i^{[2]}$ equal to $\sqrt{\lambda_i^{(3)}}$, for $i = k,l$. In 3D, the completion curve of 1D-manifolds considers two angles: the yow and pitch angles, whereas in 2D-manifolds interpolation of the two planes via the cubic spline creates the completion surface (Fig. 21).

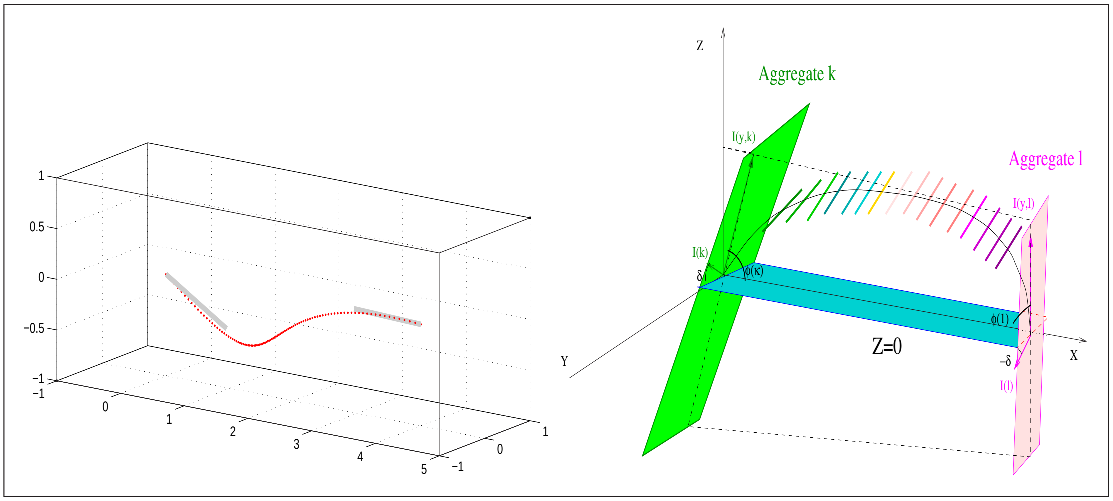

Fig. 21. **Completion of manifolds in 3D. 1D manifolds**.(left): The completion curve is drawn between the two average coordinates. **2D manifolds**. (right): Two aggregates k and l are estimated as planes. $I(k)$ and $I(l)$ are interpolated to each other through the completion curve, forming the completion surface. 2δ is the roll angle difference.

References

[1] Y. Ng Andrew, M. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, Advances in Neural Information Processing Systems 14 (2002).

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

[3] M.T. de Berg, M.J. van Kreveld, M.H. Overmars, O. Schwarzkopf, Computational Geometry: Algorithms and Applications, Springer-Verlag, Heidelberg, 2000.

[4] M. Blatt, S. Wiseman, E. Domany, Data clustering using a model granular magnet, Neural Computation 9(8) (1997) 1805–1842.

[5] A. Brandt, S. McCormick, J. Ruge. Algebraic multigrid (amg) for automatic multigrid solution with application to geodetic computations. Inst. for Computational Studies, POB 1852, Fort Colins, Colorado, 1982.

[6] A. Brandt, D. Ron, Multigrid Solvers and Multisclae Optimization Strategies, in: J.Cong, J.R. Shinnerl, (Eds.), Multilevel Optimization and VLSICAD, Kluwer, Boston, 2003, pp. 1–69.

[7] T. Cox, M. Cox, Multidimensional Scaling, Chapman and Hall, London, 1994.

[8] A. P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. of the Royal Statist. Society 39(1) (1977) 1–38.

[9] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, John Wiley and Sons, New York, 2001.

[10] B. Fischer, V. Roth, J. M. Buhmann, Clustering with the connectivity kernel, NIPS 16 (2004).

[11] B. Fischer, J. M. Buhmann, Bagging for path-based clustering, TPAMI 25(11) (2003) 1411–1415.

[12] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the Nystrom method, TPAMI 26(2) (2004) 214–225.

[13] J. H. Friedman, W. Stuetzle, Projection pursuit regression, J. Amer. Statist. Assoc. 76(376) (1981) 817–823.

[14] M. Galun, E. Sharon, R. Basri, A. Brandt, Texture segmentation by multiscale aggregation of filter responses and shape elements, ICCV, (2003) 716–723.

[15] Y. Gdalyahu, D. Weinshall, M. Werman, Self organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization, TPAMI 23(10) (2001) 1053-1074.

[16] Y. Goldschmidt, M. Galun, E. Sharon, R. Basri, A. Brandt, Fast multilevel clustering, Technical report, MCS05-09, Dept. of Computer Science and Applied Mathematics, The Weizmann Institute of Science.

[17] G. Guy, G. G. Medioni, Inference of Surfaces, 3D Curves, and Junctions From Sparse, Noisy, 3D Data, TPAMI 19(11) (1997) 1265–1277.

[18] J. A. Hartigan, M. A. Wong, A k-means clustering algorithm, App. Statistics 28 (1979) 100–108.

[19] T. Hastie, W. Stuetzle, Principal curves, J. Amer. Statist. Assoc. 84(406) (1989) 502–516.

[20] Hawkins et. al., The 2dF galaxy redshift survey correlation functions peculiar velocities and the matter density of the Universe, MNRAS 346 (2003) 1–19.

[21] A. K. Jain, R. C. Dubes, Algorithms for Clustering Data, Prentice-Hall, New Jersey, 1988.

[22] I. T. Jollife, Principal Component Analysis, Springer-Verlag, New York, 1986.

[23] M. LeBlanc, R. Tibshirani, Adaptive principal surfaces, J. Amer. Statist. Assoc. 89(425) (1994) 53–64.

[24] J. A. Peacock, Implications of 2dFGRS results on cosmic structure, AIP conference proceedings 666 (2003) 275–290.

[25] P. Peebles, The Large Scale Structure of the Universe, Princeton University Press, 1980.

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

[27] E. Sharon, A. Brandt, R. Basri, Fast multiscale image segmentation, CVPR (2000) 70–77.

[28] E. Sharon, A. Brandt, R. Basri, completion energies and scale, TPAMI 22(10) (2000) 70–77.

[29] A. Shashua, S. Ullman, Structural saliency: The detection of globally salient structures using a locally connected network, ICCV (1988) 321-327.

[30] J. Shi and J. Malik. Normalized cuts and image segmentation. TPAMI 22(8) (2000), 888-905.

[31] D. C. Stanford, A. E. Raftery, Finding curvilinear features in spatial point patterns: principal curve clustering with noise, TPAMI 22(6) (2000) 601–609.

[32] J. L. Starck, F. Murtagh, Astronomical Image and Data Analysis, Springer, New York, 2002.

[33] J. B. Tenenbaum, V. De Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[34] L. R. Williams, D. W. Jacobs, Stochastic completion fields: A neural model of illusory contour shape and salience, Neural Computation 9 (1997) 837-858.

Dan Kushnir is a PhD student at the Applied Math and CS Dept. at the Weizmann Inst. of Science. His research interests are scientific computing, geometry, pattern recognition and data analysis.

Dr. Meirav Galun is a staff scientist associated with Prof. Achi Brandt and Prof. Ronen Basri, at the Applied Math and CS Dept. at the Weizmann Inst. of Science. Her research interests are scientific computation, multiscale methods, computer vision, medical imaging and data analysis.

Prof. Achi Brandt is a professor at the Applied Math and CS Dept. in the Weizmann Institute of Science and a Professor in Residence at the Math Dept. at UCLA. He received the Landau Prize in Mathematics (1978), the Rothschild Prize in Mathematics (1990), and the Joint SIAM/ACM prize (2005). Prof. Brandt has introduced and developed multigrid and other fast multiscale computational methods in various fields of natural sciences, computer science and engineering.