Data Clustering as an Emergent Consensus of Autonomous Agents

Piotr Minakowski\textsuperscript{a}, Jan Peszek\textsuperscript{b}

\textsuperscript{a}Institute of Analysis and Numerics, Otto von Guericke University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany
\textsuperscript{b}Institute of Applied Mathematics and Mechanics, University of Warsaw, ul. Banacha 2, 02-097 Warszawa, Poland

Abstract

We present a data segmentation method based on a first-order density-induced consensus protocol. We provide a mathematically rigorous analysis of the consensus model leading to the stopping criteria of the data segmentation algorithm. To illustrate our method, the algorithm is applied to two-dimensional shape datasets and selected images from Berkeley Segmentation Dataset. The method can be seen as an augmentation of classical clustering techniques for multimodal feature space, such as DBSCAN. It showcases a curious connection between data clustering and collective behavior.

Keywords: data clustering, collective dynamics, image segmentation, differential equations

1. Introduction

The paper explores the relationship between first-order collective dynamics and data clustering, which we illustrate on color image segmentation. Our main contributions include:

- An augmentation of the DBSCAN clustering method that reduces numerical complexity.
- The technique inherits the advantages of other density-based clustering methods, with lower average numerical complexity and manageable parameters.
- Mathematically rigorous analysis of the model leading to the stopping criteria for the algorithm.
- Application of the method in color image segmentation.

Our approach is based on the recently introduced density-induced consensus protocol (DI protocol) for agent-based collective dynamics [1]. Consider \( N \) agents with \( x_i(t) \in \mathbb{R}^d \) denoting the position of \( i \)th agent in a \( d \)-dimensional space at the time \( t \geq 0 \). The agents follow the DI protocol

\[
\dot{x}_i = \kappa \sum_{k \in \mathcal{N}_i} (x_k - x_i), \quad x_i(0) = x_{i0} \in \mathbb{R}^d.
\]

Here \( \kappa > 0 \) is a fixed coupling strength whose influence amounts, in practice, to time-scaling. The neighbor set \( \mathcal{N}_i \) of \( i \)th agent is defined through the following relation: given positive parameters \( \delta \)
and $m$, for $t \geq 0$ we define,

$$k \in \mathcal{N}_i(t) \iff x_k(t) \in B(x_i(t), \delta) \text{ and } \# \{ k \in \{1, ..., N \} : x_k(t) \in B(x_i(t), \delta) \} > m,$$

where $B(x_i(t), \delta)$ is an open ball centered at $x_i(t)$ with radius $\delta$ and $\# A$ denotes the cardinal number of $A$. Thus the communication rule of the DI protocol reads as follows: the $i$th agent is influenced by $j$th agent if

- the density of agents in close proximity to the $i$th agent is substantial enough (otherwise the $i$th agent is an outlier),
- the $j$th agent is in close proximity to the $i$th agent.

Interpreting the positions $x_i$ as data points in a multimodal feature space and evolving the data in time using (1) results in a data clustering. To explain this idea, here we fix our attention to 2D data represented as vectors/positions in a Euclidean space. The positions $x_i(t) \in \mathbb{R}^2$ for $i \in \{1, ..., N \}$ evolve according to (1) eventually leading to a steady state at the time $t = \infty$, typically forming multiple clusters. Finally, the $t = \infty$ clustering is retroactively applied to the initial $t = 0$ state, to establish the segmentation, see Fig. 1.

![Figure 1](image-url)

Figure 1: An ensemble of 13 agents with visualisation of their range of interactions. A) initial positions of the agents; B) positions evolve towards the steady-state; C) steady state at $t = \infty$ defines the clusters (color coded); D) clustering applied to the initial state.

It is noteworthy that the communication rule of (1) is equivalent to the density-based spatial clustering algorithm (DBSCAN) [2] widely used in data segmentation [3]. Our method inherits the advantages (and most disadvantages) of the classical density-based clustering algorithms i.e.: no requirement of initial specification of the number of clusters, arbitrary shaped clusters, robustness to outliers and adaptability to various types of data due to the possibility of fine-tuning the parameters. The main difference is that we do not need database-oriented range-queries, which leads to the computational complexity of DBSCAN. Instead, our procedure is based on local direct connectivity defined by the neighbor sets $\mathcal{N}_i$, and thus it is less computationally demanding.
Concerning the above, the DI protocol applies to consensus dynamics as discussed in [1]. In particular, it is reminiscent of the bounded-confidence consensus model by Hegselmann and Krause [4]. Similarly, the agents are influenced only by like-minded (i.e., nearby) individuals but in the case of the DI model, the agents additionally tend to conform to group mentality and ignore the outliers. Since our method originates in consensus dynamics it can be particularly useful in segmentation of data related to collective dynamics such as the classification of pedestrians or the users of social networks. It is in itself an interesting application of non-local collective dynamics in data clustering.

The paper is organized as follows. In Section 2 we compare the DI protocol to previous research on consensus dynamics and density based clustering, focusing on unsupervised color image segmentation. In Section 3 we provide an analysis of emergent behavior of the DI protocol based on the interplay between density of the agents and connectivity. Sections 4 and 5 are dedicated to the presentation of the main segmentation algorithm and its illustration on classical data sets, respectively. Here, we also apply the results of Section 3 to derive the stopping criterion. Section 6 focuses on applications in color image segmentation. Finally, in Sections 7 and 8 we provide mathematical proofs related to the results of Section 3 and a brief conclusion to the paper, respectively.

Notation. Throughout the paper we will sometimes interchangeably refer to agents following the DI protocol as nodes, data or pixels depending on the context of applications.

2. Previous research

2.1. Consensus dynamics

Protocol [1] is a first-order variant of the DI model introduced in [1] which simulates density based alignment in the spirit of the Cucker-Smale model ([5], see also surveys [6] and [7]). First-order models similar to [1] with a variety of interaction laws have been extensively studied from the perspective of both mathematical theory and practical applications, with the prominence of applications in opinion dynamics. Opinion dynamics dates back to French’s research on social influence [8], and further, to works by De Groot [9] and Lehrer [10]. A more modern approach, including nonlinear models, was established by Krause [11] and Hegselmann jointly with Flache [12], among others. We emphasise the aforementioned work by Hegselmann and Krause [4] introducing a bounded confidence opinion dynamics model. We also recommend surveys [13] and [14] for more up-to-date information, and [15] where similar models are presented in applications to vehicular traffic and crowd dynamics.

2.2. Density based clustering

Since most of the clustering techniques [16] are application dependent, they are not tailored to the classification of arbitrary feature space. Methods that rely explicitly on the number of clusters or implicitly assume the same shape for all the clusters are not well suited to analyse data of unknown origin, see [17] for a survey.

Density based clustering is a class of unsupervised learning methods. Points that are in high-density regions in a data space are clustered together and separated from other such clusters by regions of low point density [18] [19].

As already mentioned in the introduction, we group points in a similar way as DBSCAN [2] does. The OPTICS [20] algorithm is an extension of DBSCAN and removes its dependence on parameters. Another related iterative technique is Mean-shift algorithm [21], where each object is assigned to the densest area in its vicinity, based on kernel density estimation.
Unsupervised image segmentation is an important component in many image processing systems. A nonparametric technique for the analysis of a complex multimodal feature space based on mean-shift algorithm with application to image segmentation have been presented in [22]. On the other hand, one can apply AI based approaches. The usage of convolutional neural networks for unsupervised image segmentation have been studied in [23] or [24]. Note that, image segmentation is an ill-defined problem since there is no unique ground-truth segmentation of an image against which the output of an algorithm may be compared. For the evaluation methods, we refer to [25] and survey [26].

The connection between clustering and collective dynamics, has been already explored in the context of particle swarm optimization (PSO) [27]. For density based PSO clustering algorithms we refer to [28] and [29]. Moreover, PSO based image clustering has been developed in [30]. The only similarity between PSO and our method is the agent based approach itself. However the rules of the agents’ evolution are different, and lead to different characteristics of the methods, e.g. PSO requires to a priori specify the number of clusters, and is based on solving global optimisation problem – a computationally expensive process. On the other hand, we utilise a simple rule for the evolution of agents ([1]) allowing us to employ techniques of linear consensus theory [31, 32].

3. Emergent dynamics of the DI protocol

In this section we analyse the long time dynamics of the DI protocol, focusing on cluster formation. Before we begin, let us introduce the necessary notation.

The DI consensus protocol ([1]) can be naturally represented in the language of graph theory. The ensemble \{1, ..., N\} is identified with a directed graph (digraph) (\{1, ..., N\}, \mathcal{E}) with edges \mathcal{E} defined by neighborhoods \mathcal{N}_i:

\[ \mathcal{E}(i, j) = 1 \Leftrightarrow i \in \mathcal{N}_j, \]

cf. [2]. Note that, since \(i \in \mathcal{N}_j\) does not imply \(j \in \mathcal{N}_i\), the matrix \(\mathcal{E}\) is not necessarily symmetric, and thus the graph (\{1, ..., N\}, \mathcal{E}) is directed. We shall refer to all subsets of \{1, ..., N\} as clusters. What follows, is a natural inheritance of standard notions from graph theory; for instance a cluster \(\mathcal{A} \subset \{1, ..., N\}\) is weakly/strongly connected iff its respective graph is weakly/strongly connected. We will also say that a cluster \(\mathcal{A} \subset \{1, ..., N\}\) is isolated iff it is not connected to any node outside of \(\mathcal{A}\).

Next we provide the essential notion of a densely-packed cluster introduced in [1].

**Definition 1.** We say that the cluster \(\mathcal{A} \subset \{1, ..., N\}\) is \(r\)-densely packed at the time \(t\) if

1. the set

\[ \bigcup_{i \in \mathcal{A}} B(x_i(t), r/2) \]

is connected,

2. each open ball \(B(x_i(t), r)\), for \(i \in \mathcal{A}\) contains more than \(m\) agents.

There are multiple connectivity- and graph-related implications of Definition 1. Most of them, are presented in Section 7 Lemma [4]. The main application is the following theorem (also proved in Section [7] on sufficient conditions ensuring that the clusters collapse into a single steady state.
Theorem 1. Suppose that at any time \( t_0 \) the ensemble \( \{1, ..., N\} \) consists of \( K \) isolated connected clusters \( \mathcal{A}_1, ..., \mathcal{A}_K \) with the property that the convex hulls of \( \delta/2 \)-neighborhoods of the clusters do not intersect, i.e.

\[
\text{conv} \left( \bigcup_{i \in \mathcal{A}_k} B(x_i, \delta/2) \right) \cap \text{conv} \left( \bigcup_{i \in \mathcal{A}_l} B(x_i, \delta/2) \right) = \emptyset
\]

(3)

for all \( k, l \in \{1, ..., K\} \). Then (3) persists in time and the clusters remain isolated. Moreover, each \( r \)-densely packed cluster \( \mathcal{A} \) with \( r \) satisfying

\[
\frac{r}{\delta} \frac{Z^3}{6\#A} \leq \frac{m}{\#A} Z, \quad Z := e^{\frac{2m}{3(\#A)^3}}
\]

(4)

is \( \delta \)-densely packed indefinitely, and converges to the steady state

\[ x_A := \frac{1}{\#A} \sum_{i \in \mathcal{A}} x_i. \]

Remark 1. Condition (4) may not be clear at the first glance, but is easily viable from the perspective of numerical computations. Intuition behind it is as follows. If \( \mathcal{A} \) is \( r \)-densely packed with \( r \leq \delta \), then it is strongly connected (see Lemma 1 in Section 7 below). Consequently, if \( r \) is significantly smaller than \( \delta \), the agents have room to move around, before the clusters cease to be at least \( \delta \)-densely packed. Thus, the quantity \( \frac{r}{\delta} \frac{Z^3}{6\#A} \) represents the cluster’s rigidity, i.e. small \( \frac{r}{\delta} \) means that the cluster is flexible and the agents may move a lot before connectivity breaks. Condition (4) requires rigidity to be small compared to quantities related to the cluster’s volume and the algebraic connectivity for (1).

In practice we usually have \( 2m \ll 3(\#A)^3 \), for which

\[ Z \approx 1 \]

and then (4) can be reduced to

\[
\frac{r}{\delta} \frac{Z^3}{6\#A} \approx \frac{\delta m}{6\#A}.
\]

(5)

We use simplification (5) in numerical computation.

Remark 2. Theorem 1 states that as soon as convex hulls of the clusters’ influence regions are disjoint and any cluster is sufficiently densely packed, then, from the perspective of large-time behaviour, that cluster can be immediately replaced by its center of mass \( x_A \). It naturally leads to a stopping criterion for the clustering algorithm presented in Section 4.1 below.

4. Algorithm

The continuous model presented in Section 1 can be simulated with a time-discretization scheme. This results in an iterative procedure that, for a given set of agents, performs clustering by advancing their positions in time up to the point when a sufficiently densely packed configuration is reached. The procedure is presented in Algorithm 1 with a detailed description of the key steps in sections 4.1 - 4.8.
Algorithm 1: Segmentation Procedure.

**Input:** data(0) and parameters ($\delta$, $m$, $n_{\text{max}}$)

Pre-processing (optional);
Build $\delta$-lattice;

for $n = 0; n < n_{\text{max}}; n++$ do

Interactions(data($n$), $\delta$, $m$);
Advance the data in time;

end
Build $\epsilon$-lattice;
Identify clusters;
Assign colors and outliers to clusters (optional);
Post-processing (optional);

4.1. Input

The input consists of the given data(0) scaled to the unit cube $[0,1]^d$, parameters $\delta$, $m$, and the stopping time $n_{\text{max}}$. Parameters $\delta$ and $m$ are chosen to fit particular types of data, similar to other unsupervised data segmentation methods. The influence of parameters is investigated in Section 6.1.

In order to fix the stopping time $n_{\text{max}}$ we test Algorithm 1 on sample sets of data. We perform the loop from Algorithm 1 until all clusters satisfy (3) and are $r$-densely packed with $r$ satisfying (5). Then we set $n_{\text{max}} = \text{the number of the last iteration}$. This process is computationally expensive, but it needs to be performed only once for each type of similar data. For example, in all applications presented in the sequel, we take $n_{\text{max}} = 10$.

4.2. Build $\delta$-lattice

By (1) and (2) the communication protocol is local and depends on interaction range $\delta$. Therefore, we introduce the division of $[0,1]^d$ into a regular lattice of $N_d$ cells, with the edge length $N_l = 1/\delta$. This serves the purpose of reducing the complexity of neighbor queries, in order to compute the right hand side of (1). The agents within $\delta$-distance are to be found in adjacent cells. Thus, for uniformly distributed data, the complexity scales linearly with the number of agents. This method is applied, for instance, in particle simulations of liquids, c.f. [33].

4.3. Interactions

For a fixed $n \in \{0,\ldots,n_{\text{max}}\}$, we establish the connectivity between the agents at the $n$th iteration, i.e. the neighbor sets $\mathcal{N}_i(n)$. It suffices to check conditions (2) for each $i \in \{1,\ldots,N\}$ and each $k$ belonging to the same $\delta$-lattice cell as $i$ or to the adjacent cells.

4.4. Advance the data in time

For temporal discretization we use a scheme from the Runge–Kutta family of explicit methods. We advance in time with a well-known explicit Euler method

\[
x_{i}^{n+1} = x_{i}^{n} + (\Delta t) \kappa \sum_{k \in \mathcal{N}_i(n)} (x_{k}^{n} - x_{i}^{n}), \quad x_{i}^{0} = x_{i0}.
\]
Here $N_i(n)$ is the neighbor set obtained in the previous step. We take $(\Delta t)\kappa = 1/K_n$, where

$$K_n = \max_i \left\{ \max \{ \#N_i(n), m \} \right\}.$$

Naturally, other choices of $(\Delta t)\kappa$ are possible, but one should make sure that the time-scale is small, whenever the maximal local density of the agents is large.

### 4.5. Identify clusters

After completing the $n_{\text{max}}$ iteration we construct an $\epsilon$-lattice, for some $\epsilon \leq \delta$. Note that, the maximal iteration $n_{\text{max}}$ has been chosen so that we expect all isolated connected clusters to be $r$-densely packed with $r$ satisfying (5). In such a case the $\epsilon$-lattice cells emulate $r$-densely packed clusters. The cluster assignment is performed as follows:

1. Cells of the $\epsilon$-lattice are divided into 2 groups: core cells with more than $m$ agents inside and outlier cells with $m$ or less agents inside.
2. Any core cell forms a cluster along with all of adjacent core cells and, transitively, all subsequent adjacent core cells etc..
3. Agents belonging to core cells are then assigned to the respective clusters.
4. Agents belonging to the outlier cells become outliers and are considered noise.

Finally the cluster assignment is retroactively applied to the initial values i.e. to data(0), as showcased in Fig. 1.

**Remark 3.** The scale $\epsilon > 0$ of the $\epsilon$-lattice chosen to be much greater than the density threshold (5), i.e. $r \ll \epsilon$. However, at $n_{\text{max}}$ we expect all of the connected isolated clusters to be $r$-densely packed and asymptotically stable, by Theorem (7). Therefore the only inadequacy introduced by the procedure described in Section 4.5 compared to what we would obtain with a finer $r$-lattice, amounts to the merging of clusters. The reasonable choice of $\epsilon$ is in the same order of magnitude as $\delta$. In the forthcoming examples we take either $\epsilon = \frac{\delta}{2}$ or $\epsilon = \frac{\delta}{4}$, since these values provide a good balance between numerical complexity, accuracy of the algorithm and resistance to the curse of dimensionality.

### 4.6. Assign colors and outliers to clusters

Optionally, for instance in the applications to color image segmentation, at this point we assign values (colors) to each of the clusters obtained in Step 4.5. To do so, we simply take the average value of the data within each cluster. Thus, if $\mathcal{A}$ is one of the clusters, it is assigned with the average value $x_{\mathcal{A}} = \sum_{i \in \mathcal{A}} x_i / \# \mathcal{A}$. Then, each outlier cell (and the agents within) is assigned to the cluster with the closest average value $x_{\mathcal{A}}$.

In the case of color image segmentation, this step is responsible for defining the color scheme and dealing with the outliers.
4.7. Complexity

For each time-step the most computationally expensive part of the algorithm is to establish the connectivity between the agents described in Section 4.3. Taking advantage of the $\delta$-lattice structure, for each agent we need to examine neighbors in adjacent lattice cells. In that case, only a limited number of agents must be visited multiple times, therefore the average complexity of $O(N)$ is obtained. Moreover, we perform constant number of iterations, since $n_{\text{max}} \ll N = \text{number of agents}$.

As described in Section 4.5, identification of clusters is based on the $\epsilon$-lattice. We denote by $k$ the number of core cells in the $\epsilon$-lattice. Consequently, identifying clusters procedure requires $O(k \log k)$ operations. Therefore an overall average runtime complexity of $O(N) + O(k \log k)$ is obtained. If

$$k \log k \leq N, \quad (6)$$

then we obtain the complexity of order $O(N)$. Thanks to the fact that we only need to work with core cells, we can estimate $k$ in terms of $\delta$, regardless of the choice of $\epsilon$. By the definition of the core cells, $km \leq N$ and assuming uniform distribution of the agents, i.e. $m_{\delta} \approx N$ leads to $k \leq \frac{1}{\delta}$. Thus (6) is ensured if

$$\frac{1}{\delta^d} \log \frac{1}{\delta^d} \leq N. \quad (7)$$

This is the key point of our contribution, namely the development of a method of low complexity, c.f. [34]. It outperforms mid complexity methods, e.g. DBSCAN, whose complexity is driven by distance queries. Even if an indexing structure is used, a neighbourhood query executes in $O(\log N)$ leading to an overall average complexity of DBSCAN of order $O(N \log N)$, with the worst case of $O(N^2)$, c.f. [35].

**Remark 4.** The improvement of the algorithm’s numerical complexity hinges on assumption (6), which we achieve by ensuring that $\delta$ satisfies (7) for any fixed $N$. It creates a lower bound $\delta_0(N,d)$ for the set of admissible parameters $\delta$.

Observe, that taking $n_{\text{max}} = 0$, i.e. immediately skipping to step 4.5 of the algorithm, we actually perform a simplified variant of the DBSCAN segmentation with $\epsilon$ as the neighborhood size. However, in a high-dimensional case $\delta_0(N,d)$ can be relatively large resulting in far too few clusters. In other words, proper clustering with the DBSCAN algorithm requires $\delta$ to be small enough, so that we cannot ensure (6) and we do not gain much in terms of numerical complexity. We visualize this phenomenon in the case of color image segmentation in Fig. [3]. Indeed, the image in column C and row 1 presents such a scenario: Algorithm 1 performed with $n_{\text{max}} = 0$. It leads to the emergence of only 1 cluster and a poor representation of the original image in column A and row 1. It indicates that the key point of our method, i.e. the evolution of the agents’ positions in time, which breaks large clusters, is essential.

4.8. Pre- and post processing

In order to improve clustering, we may perform pre- or post-processing. For instance, in color image segmentation, we pre-process images with Gaussian blur. Post-processing might be needed if the resulting segmentation produces a fairly large number of clusters. One can merge clusters or run Identify clusters with a lattice of larger scale.
5. Clustering of Two-dimensional Exemplary Datasets

In this section, we present a comparison between our DI clustering and two classical unsupervised methods: mean-shift and DBSCAN. Each row of Fig. 2 presents an example of shape dataset, see [36] and references therein. In the first column we see the ground-truth clustering. In the second and third – mean-shift and DBSCAN, respectively. The remaining columns are dedicated to the DI protocol.

The parameters for the applied algorithms are as follows $m = 1$, $\delta = 0.1$, $\epsilon = \frac{\delta}{2} = 0.05$ and $n_{\text{max}} = 40$ DBSCAN is run with the parameters $\delta$ and $m$ that are the same as in DI protocol. Mean-shift uses default values and bandwidth estimation, see [37] for details.

In the forth column of Fig. 2 pixels/agent are in position after the data was advanced $n_0$ steps in time. In Algorithm 1, it corresponds to the situation at the end of the loop. Then the fifth column presents the situation after building the $\epsilon$-lattice and assigning clusters within the lattice. Finally, column six represents the clustering applied to the original datasets.

Note that the parameters used in Fig. 2 were not particularly chosen to suit the DI protocol at the expense of other algorithms. The conclusion related to the accuracy of the algorithm (and
disregarding numerical complexity) that arises upon examination of Fig. 2 and similar pictures is as follows:

- The DI protocol, similar to the DBSCAN algorithm and contrary to the mean-shift, is suited to cluster data with arbitrary shapes.

- The DI protocol tends to split clusters, which in many cases prove useful, e.g. example 3 and 4 and in other times can be detrimental, e.g. example 2 and 5. This is further visualized and discussed in Fig. 3 in the next section.

- The advantage of the DI protocol lies in simplification of data and not necessarily clustering itself. Particularly, columns 4 and 5 present a simplification of the original ground truth in the first column. The general shape of the data is retained but the distributions are less complex. This may prove useful as a method to reduce the dimension of the data.

6. Application to Color Image Segmentation

In order to further illustrate our method, we apply Algorithm 1 to selected pictures from Berkeley Segmentation Dataset (BSD) [35]. Size of the images is 481 times 321 what results in $N = 154401$ pixels.

The position $x_i(t) \in \mathbb{R}^5$ represents the $i$th pixel in a 5D feature space, where the modality is realised by combining 2D spatial positions of pixels in an image and their 3D representations in a color space (e.g. in RGB). In the feature space we define the combined distance $d(i,j)$ between pixels $i$ and $j$, the spatial distance $d_s(i,j)$ and color distance $d_c(i,j)$ as follows:

$$
d_s(i,j)^2 = (y_i^1 - y_j^1)^2 + (y_i^2 - y_j^2)^2,
$$

$$
d_c(i,j)^2 = (r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2,
$$

$$
d(i,j) = \sqrt{d_s(i,j)^2 + d_c(i,j)^2},
$$

where $y^1$ and $y^2$ are spatial coordinates and $r$, $g$, $b$ are color values in the RGB color space. All variables are normalised to $[0,1]$, therefore we work in a 5D unit cube.

Segmentation follows Algorithm 1. We apply Gaussian blur as pre-processing and for post-processing we assign colors and outliers as described in Section 4.6. This process is illustrated in Figure 3.

Column A of Figure 3 consists of 4 images with original positions of the pixels and colors inherited from 0, 3, 7 and 10 iteration in each row, respectively. For instance, the image in column A and row 2 presents the 5D pixels with combined coordinates $x^A_i(3) = (y^1_i(0), y^2_i(0), r_i(3), g_i(3), b_i(3))$. Column B represents the projection of each pixel $x^A_i$ onto the 3D RGB space, e.g. for $x^A_i(3)$ we have $(r_i(3), g_i(3), b_i(3))$ in the image in column B and row 2. Column C consists of images with original positions and colors obtained through the application of the cluster identification steps of Algorithm 1 (c.f. Sections 4.5 and 4.6) with $n_{max} = 0, 3, 7, 10$, respectively. Column D represents positions of clusters in the RGB space.

Observe the interesting case of the image in Column C and row 1. Since no iterations of the loop in Algorithm 1 was performed, the pixels are not sufficiently densely packed and only one cluster, say $A_1$, is identified by the procedure in Section 4.5. Then, according to Section 4.6, the color of each pixel in $A_1$ is changed to the average color of all pixels in $A_1$, which happens to be blue. All remaining pixels outside of $A_1$ are outliers and are then assigned the same blue color.
In order to show the parameters’ influence, we apply Algorithm 1 with various \( m \) and \( \delta \) to a picture of peppers, Figure 4.

We pick the interaction range \( \delta \in \{0.1, 0.2\} \) and \( \epsilon = \frac{\delta}{4} \). This choice is connected to a number of pixels \( N \) and the dimension \( d \), \( \delta \approx \frac{3}{\sqrt{N}} \approx 0.1 \). This corresponds to the length of an edge of a 5-dimensional cube with volume \( 1/N \) (average volume for one pixel).

The parameter \( m \) is set to be equivalent to the minimal cluster density being \( \eta \) times greater than an average pixel density, \( m = \frac{\delta}{15} \pi^2 \eta N \delta^5 \), with \( \eta \in \{1,5\} \).

6.1. Parameters

Figure 5 presents: fully segmented image (Column C) with \( n_{\text{max}} = 10 \), and two projections onto the 3D \( RGB \) space (Columns B and D), as described above for Figure 1. We observe that the number of clusters is directly connected to the interaction range \( \delta \), but seems mostly independent of parameter \( m \). Instead, parameter \( m \), drives the dynamics of agents, c.f. Column B, and thus –
the final cluster assignment. Indeed, comparing rows 1 and 2 in Fig. 5 we observe that the clusters in Column D are almost the same, but the segmented images in Column C differ.

Note, that in extreme cases, for large $\delta$ and small $m$, one will obtain only one cluster. On the other hand, with small $\delta$ and large $m$ each point will be placed into its cluster, like in the input image.

B) $\delta = 0.2, m = 260, \eta = 1$

C) $\delta = 0.2, m = 1300, \eta = 5$

D) $\delta = 0.1, m = 8, \eta = 1$

Unsupervised color image segmentation is an image processing task without a unique result. The BSD provides several human-performed segmentations for each picture. As pointed out in [26] one cannot guarantee that any manually-generated segmentation image is better than another. This is especially the case with nature images. Thus, the dependence on parameters can be seen.
as an advantage of the presented method, allowing to encapsulate different results of color image segmentation, similar to the manually-generated cases.

Note that the provided parameters are closely related to the average density of pixels. In practice, taking $\delta$ and $m$ from small ranges of values leads to relatively small differences in the resulting segmented pictures, which corresponds to the slight variation between human-generated segmentations. Thus, restriction of the set of admissible parameters to values related to average density of the pixels, while ensuring that bounds established in Section 4.7 are satisfied, makes finding the proper values of parameters manageable. This is further showcased by Fig. 6 below, where multiple different pictures are segmented using the presented method with the same values of parameters leading to mostly reasonable results.

6.2. Segmentation Results on BSDS500

Next, we present segmentation results on selected images from BSD, see Fig. 6. Images are from both the test and the train set. Each pair consists of the original image on the left and the segmented image on the right. Parameters are $\delta = 0.15$, $m = 308$, $\epsilon = \frac{\delta}{2}$, $\eta = 5$, and $n_{\text{max}} = 10$. We recall that $m = \frac{2}{15} \pi^2 \eta N \delta^3$. The selected interaction range results in a relatively small number of clusters. Note that, it depends on the type of picture, typically images of nature exhibit similar colour, resulting in fewer clusters.

![Figure 6: Additional segmentation results on the BSD.](image)

Edge detection

Finally, we briefly showcase the impact of our method on segmentation on contour detection. In Fig. 7 we present, image and edge detection for input (left) and segmented (right) image, respectively. Contours are detected by the method of Canny. We observe, the improvement of edge detection for the segmented image compared to the direct application of Canny method to the original image.
7. Mathematical proofs

In this section we prove Theorem 1. The starting point is Lemma 1 which connects the notion of \( r \)-densely packed clusters with their connectivity. We recall the notation established in Section 3.

Throughout this section we denote
\[
\mathbf{x} := (x_1, \ldots, x_N) \in \mathbb{R}^{dN}.
\]

**Lemma 1.** Suppose that \( \mathbf{x} \) is a smooth solution to (1) with constant coefficients in \([t_0, t_1)\). Assume further that \( \mathcal{A} \) is an isolated \( r \)-densely packed cluster at each \( t \in [t_0, t_1) \) for \( r \leq \delta \). Then

1. a cluster \( \mathcal{A} \) treated as a graph is strongly connected and undirected,

2. The graph diameter \( d_{\mathcal{A}} \) of cluster \( \mathcal{A} \), i.e. the maximal shortest path between any two nodes, is upper-bounded by
   \[
   d_{\mathcal{A}} \leq \begin{cases} 
   \frac{\# \mathcal{A}}{m+1} - 1 & \text{if } m \geq 2, \\
   1 & \text{if } m = 1. 
   \end{cases}
   \tag{8}
   \]

3. The spatial diameter \( D_{\mathcal{A}} := \max\{|x_i - x_j| : i, j \in \mathcal{A}\} \) of a cluster \( \mathcal{A} \) is upper-bounded by
   \[
   D_{\mathcal{A}} \leq r d_{\mathcal{A}}.
   \]

4. There exists \( \lambda > 0 \) such that
   \[
   |x_i(t) - x_{\mathcal{A}}| \leq e^{-\lambda(t-t_0)}|x_i(t_0) - x_{\mathcal{A}}|, \quad \forall i \in \mathcal{A}, \ t \in [t_0, t_1),
   \]
   where \( x_{\mathcal{A}} \) is the center of mass of \( \mathcal{A} \). Moreover the exponent \( \lambda \) satisfies
   \[
   \lambda \geq \frac{4\kappa}{d_{\mathcal{A}} \# \mathcal{A}}. \tag{9}
   \]
Proof. We refer to [11] Lemma 1 and Lemma 2, where variants of assertions 1 and 4 are proved by modifying the theory of linear consensus developed in [31] and [32]. The only difference here is that cluster $\mathcal{A}$ is undirected due to the symmetry of weight in (1), which further implies that the steady state $x_\mathcal{A}$ is indeed the center of mass of $\mathcal{A}$. Assertion 1 implies that $\mathcal{A}$ is an isolated, connected undirected graph with a minimal degree of $m$ and assertion 2 holds true due to Erdős, Pach, Pollack and Tuza’s classical work [10, Theorem 1]. To prove assertion 3, consider the graph $\mathcal{G} := (\mathcal{A}, \mathcal{E})$, with $\mathcal{E}(i,j) = 1$ if $\mathcal{E}(i,j) = 1$ and $|x_i - x_j| \leq r$. Then $\mathcal{G}$ is an $r$-densely packed subgraph of $(\mathcal{A}, \mathcal{E})$ with the same nodes and fewer edges. Any two nodes $i$ and $j$ in $\mathcal{A}$ are connected by a path $\pi(i,j) \in \mathcal{E}$, represented by a sequence of nodes $i = k_1, ..., k_l = j$ with

$$l \leq d_{\mathcal{A}} \text{ and } |x_{k_\alpha} - x_{k_{\alpha+1}}| \leq r \text{ for all } \alpha \in \{1, ..., l - 1\}.$$ Consequently, $|x_i - x_j| \leq rd_{\mathcal{A}}$.

Finally, since $\lambda(t)$ is the algebraic connectivity of the graph $(\mathcal{A}, \mathcal{E})$ multiplied by $\kappa$, inequality (9) follows from McKay’s result published by Mohar in [11, Theorem 4.2], which states that the algebraic connectivity of the graph is lower-bounded by $4$ divided by the graphs diameter and its number of nodes.

We proceed with the following useful proposition.

Proposition 1. For any smooth solution $x$ to system (1) and all $t \geq t_0 \geq 0$ we have

$$\mathcal{C}(t) := \text{conv}\{x_i(t) : i \in \{1, ..., N\}\} \subset \mathcal{C}(t_0).$$

Moreover the maximal velocity of each agent is uniformly bounded, i.e.

$$V_i(t) := \max_{t \geq t_0} |\dot{x}_i(t)| \leq \kappa \# \mathcal{N}_i \delta.$$ (11)

Proof. Fix any $s \geq t_0$ and let $x_i(s)$ belong to the boundary of $\mathcal{C}(s)$ and assume without a loss of generality that $x_i(s) = 0$. It suffices to show that $\dot{x}_i(s)$ belongs to the cone $\bigcup_{l \geq 0} \tau \mathcal{C}(s)$. With $\chi \{k \in \mathcal{N}_i\}$ denoting the characteristic function of the event that $k \in \mathcal{N}_i(t)$, we have

$$\dot{x}_i(s) = \kappa \sum_{k \in \mathcal{N}_i} (x_k(s) - x_i(s))$$

$$= \kappa \# \mathcal{N}_i \sum_{k=1}^{N} \frac{\chi \{k \in \mathcal{N}_i\}}{\# \mathcal{N}_i} x_k(s) \in \kappa \# \mathcal{N}_i \mathcal{C}(s),$$ (12)

since the sum on the right-hand side of (12) is a convex combination of elements belonging to $\mathcal{C}(s)$.

The proof of (11) follows immediately from (1) and (2) after noting that $|x_i - x_k| \leq \delta$ for $k \in \mathcal{N}_i$.

The following lemma is the essential part of the proof of Theorem 1.

Lemma 2. Let $x$ be a smooth solution to (1) and let $\mathcal{A}$ be a cluster in $\{1, ..., N\}$ that remains isolated at all times. Then there exists $r^* < \delta$ such that if $\mathcal{A}$ is $r$-densely packed with $r \leq r^*$, then $\mathcal{A}$ is at least $\delta$-densely packed for all $t \geq t_0$. 

15
Proof. Any $r$-densely packed cluster is $r^*$-densely packed whenever $r \leq r^*$, hence for the remainder of the proof we will assume that $r = r^*$. Since, at $t = t_0$, the agents are $r$-densely packed for $r < \delta$ and the maximal velocity of the agents is uniformly bounded (see Proposition 1), there exists a time interval $[t_0, T)$, such that the agents are $\delta$-densely packed for $t \geq t_0$.

Cluster $\mathcal{A}$ is isolated and system (1) is piecewise linear with finitely many possible right-hand sides, and thus, Lemma 1 holds locally at each $t \geq t_0$. Therefore there exists a piecewise constant exponent $\lambda(t) > 0$ as in assertion 4 of Lemma 1. By assertion 2 as well as inequality (9) from Lemma 1, the exponent has the uniform lower bound

$$\lambda(t) \geq \frac{4\kappa}{d_A \# \mathcal{A}} \geq \frac{4\kappa m}{3(\# \mathcal{A})^2} =: \lambda_+.$$ 

Therefore for all $t \geq t_0$ and all $i \in \mathcal{A}$ we have

$$|x_i(t) - x_A(t)| \leq e^{-\lambda_+(t-t_0)}D_A(t_0),$$

$$D_A(s) := \sup_{i,j \in \mathcal{A}} |x_i(s) - x_j(s)|.$$ (13)

Let

$$T := \sup\{t > t_0 : \mathcal{A} \text{ is } \delta\text{-densely packed in } [t_0, t)\} > 0.$$ 

We shall chose $r$ small enough so that $T = \infty$. For all $i, j \in \mathcal{A}$ and all $t \in [t_0, T)$, we have

$$|x_i(t) - x_j(t)| \leq |x_i(t) - x_A(t)| + |x_j(t) - x_A(t)|.$$ (14)

Let us fix in (14) any pair $(i, j)$ such that $|x_i(t_0) - x_j(t_0)| \leq r$. Such pairs exist since the ensemble is $r$-densely packed initially. Applying (13) to the right-hand side of (14) leads to

$$|x_i(t) - x_j(t)| \leq 2e^{-\lambda_+(t-t_0)}D_A(t_0).$$ (15)

Alternatively $|x_i(t) - x_j(t)|$ can be upper-bounded using equation (1) and Proposition 1 yielding

$$|x_i(t) - x_j(t)| \leq |x_i(t) - x_i(t_0)| + |x_j(t) - x_j(t_0)|$$

$$+ |x_i(t_0) - x_j(t_0)| \leq 2\kappa \# \mathcal{A} \delta (t-t_0) + r,$$

which ensures that

$$|x_i(t) - x_j(t)| < \delta$$

at least as long as

$$t - t_0 < \frac{1}{2\kappa \# \mathcal{A}} \left(1 - \frac{r}{\delta}\right).$$

However for $t - t_0 \geq \frac{1}{2\kappa \# \mathcal{A}} \left(1 - \frac{r}{\delta}\right)$, by (15) we have

$$|x_i(t) - x_j(t)| \leq 2e^{-\lambda_+(1-t_0)}D_A(t_0)$$

16
and we require
\[ \tilde{r} := 2e^{-\frac{\lambda}{2(\#A)} (1-\frac{\varepsilon}{\delta})} D_A(t_0) < \delta. \]  
(16)

Then assertions 2 and 3 of Lemma 1 imply that
\[ D_A < 3r \frac{\#A}{m}. \]

Combining the above bounds yields
\[ \tilde{r} < 6re^{-\frac{2m}{3(\#A)} (1-\frac{\varepsilon}{\delta})} \frac{\#A}{m} \leq \delta, \]
which can be further rearranged into (4). Condition (4) holds for any sufficiently small \( r > 0 \).

In conclusion, any pair \( i \) and \( j \), initially of distance at most \( r \) from each other, remains of distance \( \tilde{r} < \delta \) from each other throughout \([t_0, T)\). Thus \( A \) is \( \tilde{r} \)-densely packed in \([t_0, T)\) with \( \tilde{r} < \delta \) and, by continuity, it is at least \( \frac{\varepsilon+\delta}{2} \)-densely packed at \( t = T \). Then, exactly as at the beginning of the proof, \( A \) remains at least \( \delta \)-densely packed past the time \( T \) which stands in contradiction with the definition of \( T \), unless \( T = \infty \). The proof is finished.

Lemma 2 ensures that \( A \) remains \( \delta \)-densely packed as long as (4) holds and \( A \) is an isolated cluster. We are now ready to finalise the proof of Theorem 1.

Proof of Theorem 1: First observe that, by Proposition 1, condition (3) persists in time. Hence, each of the clusters \( A_1, \ldots, A_K \) is isolated, and thus Lemma 2 applies with \( r^* \) satisfying condition (4). Therefore, each \( r^*-\)densely packed cluster is at least \( \delta \)-densely packed indefinitely. Consequently, for such clusters, assumptions of Lemma 1 are satisfied, and inequality (13) holds for all \( t \geq t_0 \) thereby ensuring the convergence in Theorem 1. The proof is finished.

8. Conclusions and Outlook

The paper explores the relation between density-based data segmentation and collective behavior, augmenting the DBSCAN, by letting the data evolve in accordance with a non-local interaction law. We provided a rigorous mathematical foundation for the method as well as an illustration in the case of color image segmentation. Various aspects of the algorithm are discussed: the influence of its parameters, stopping time and numerical complexity. In particular, we achieve a low \( O(N) \) average numerical complexity with parameters that lead to the emergence of an appropriate number of clusters. The key point is the observation that an evolution of the data according to a density-based first-order ODE system has a low computational cost, and breaks unreasonably large clusters that would be obtained using other methods such as DBSCAN.

Future research will be dedicated to the connections between collective dynamics and more modern clustering techniques.

Acknowledgment

PM acknowledges the financial support by the Federal Ministry of Education and Research of Germany, grant number 05M16NMA and the support of the GRK 2297 MathCoRe, funded by the Deutsche Forschungsgemeinschaft, grant number 314838170. JP was partially supported by the Polish Narodowe Centrum Nauki grant No. 2018/30/M/ST1/00340 (HARMONIA).
References

[1] P. Minakowski, P. B. Mucha, J. Peszek, Density-induced consensus protocol, Mathematical Models and Methods in Applied Sciences (Jul. 2020). doi:10.1142/s0218202520500451

[2] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD’96, AAAI Press, 1996, p. 226–231.

[3] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, X. Xu, Dbscan revisited, revisited: Why and how you should (still) use dbscan 42 (3) (Jul. 2017). doi:10.1145/3068335.

[4] R. Hegselmann, U. Krause, Opinion dynamics and bounded confidence: Models, analysis and simulation, Journal of Artificial Societies and Social Simulation 5 (2002) 1–24.

[5] F. Cucker, S. Smale, Emergent behavior in flocks, IEEE Trans. Automat. Control 52 (5) (2007) 852–862. doi:10.1109/TAC.2007.895842.

[6] Y.-P. Choi, S.-Y. Ha, Z. Li, Emergent dynamics of the Cucker-Smale flocking model and its variants, in: Active particles. Vol. 1. Advances in theory, models, and applications, Model. Simul. Sci. Eng. Technol., Birkhäuser/Springer, Cham, 2017, pp. 299–331.

[7] P. Minakowski, P. B. Mucha, J. Peszek, E. Zatorska, Singular Cucker–Smale Dynamics, Springer International Publishing, Cham, 2019, pp. 201–243. doi:10.1007/978-3-030-20297-2_7.

[8] J. R. P. French, A formal theory of social power, Psychological Review 63 (3) (1956) 181–194. doi:10.1037/h0046123.

[9] M. H. Degroot, Reaching a consensus, Journal of the American Statistical Association 69 (345) (1974) 118–121. doi:10.1080/01621459.1974.10480137.

[10] K. Lehrer, Social consensus and rational agnoiology, Synthese 31 (1) (1975) 141–160. URL http://www.jstor.org/stable/20115060

[11] U. Krause, A discrete nonlinear and non-autonomous model of consensus formation, in: Communications in difference equations (Poznan, 1998), Gordon and Breach, Amsterdam, 2000, pp. 227–236.

[12] A. Flache, R. Hegselmann, Understanding Complex Social Dynamics: a Plea for Cellular Automata Based Modelling, Journal of Artificial Societies and Social Simulation 1 (3) (1998) 1–1. URL https://ideas.repec.org/a/jas/jasssj/1998-5-1.html

[13] G. Albi, L. Pareschi, G. Toscani, M. Zanella, Recent advances in opinion modeling: control and social influence, in: Active particles. Vol. 1. Advances in theory, models, and applications, Model. Simul. Sci. Eng. Technol., Birkhäuser/Springer, Cham, 2017, pp. 49–98.

[14] A. Aydoğdu, M. Caponigro, S. McQuade, B. Piccoli, N. Pouradier Duteil, F. Rossi, E. Trélat, Interaction network, state space, and control in social dynamics, in: Active particles. Vol. 1. Advances in theory, models, and applications, Model. Simul. Sci. Eng. Technol., Birkhäuser/Springer, Cham, 2017, pp. 99–140.
[15] G. Albi, N. Bellomo, L. Fermo, S.-Y. Ha, J. Kim, L. Pareschi, D. Poyato, J. Soler, Vehicular traffic, crowds, and swarms: from kinetic theory and multiscale methods to applications and research perspectives, Math. Models Methods Appl. Sci. 29 (10) (2019) 1901–2005. doi:10.1142/S0218202519500374

[16] C. C. Aggarwal, C. K. Reddy, Data Clustering: Algorithms and Applications, 1st Edition, Chapman & Hall/CRC, 2013.

[17] A. K. Jain, R. P. W. Duin, J. Mao, Statistical pattern recognition: A review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37. doi:10.1109/34.824819

[18] M. Herbin, N. Bonnet, P. Vautrot, A clustering method based on the estimation of the probability density function and on the skeleton by influence zones. application to image processing, Pattern Recognition Letters 17 (11) (1996) 1141–1150. doi:10.1016/0167-8655(96)00085-2

[19] H.-P. Kriegel, P. Kröger, J. Sander, A. Zimek, Density-based clustering, WIREs Data Mining and Knowledge Discovery 1 (3) (2011) 231–240. doi:10.1002/widm.30

[20] M. Ankerst, M. M. Breunig, H.-P. Kriegel, J. Sander, OPTICS, ACM SIGMOD Record 28 (2) (1999) 49–60. doi:10.1145/304181.304187

[21] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (8) (1995) 790–799.

[22] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619. doi:10.1109/34.1000236

[23] A. Kanezaki, Unsupervised image segmentation by backpropagation, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018. doi:10.1109/icassp.2018.8462533

[24] W. Kim, A. Kanezaki, M. Tanaka, Unsupervised learning of image segmentation based on differentiable feature clustering, IEEE Transactions on Image Processing 29 (2020) 8055–8068. doi:10.1109/tip.2020.3011269

[25] R. Unnikrishnan, C. Pantofaru, M. Hebert, Toward objective evaluation of image segmentation algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 929–944. doi:10.1109/TPAMI.2007.1046

[26] H. Zhang, J. E. Fritts, S. A. Goldman, Image segmentation evaluation: A survey of unsupervised methods, Computer Vision and Image Understanding 110 (2) (2008) 260 – 280. doi:https://doi.org/10.1016/j.cviu.2007.08.003

[27] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE. doi:10.1109/mhs.1995.494215

[28] M. Alswaiti, M. Albughldadi, N. A. M. Isa, Density-based particle swarm optimization algorithm for data clustering, Expert Systems with Applications 91 (2018) 170–186. doi:10.1016/j.eswa.2017.08.050
[29] C. Guan, K. K. F. Yuen, F. Coenen, Particle swarm optimized density-based clustering and classification: Supervised and unsupervised learning approaches, Swarm and Evolutionary Computation 44 (2019) 876–896. doi:10.1016/j.swevo.2018.09.008

[30] M. Omran, A. P. Engelbrecht, A. Salman, Particle swarm optimization method for image clustering, International Journal of Pattern Recognition and Artificial Intelligence 19 (03) (2005) 297–321. doi:10.1142/s0218001405004083

[31] R. Olfati-Saber, Flocking for multi-agent dynamic systems: algorithms and theory, IEEE Trans. Automat. Control 51 (3) (2006) 401–420. doi:10.1109/TAC.2005.864190

[32] R. Olfati-Saber, R. M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Trans. Automat. Control 49 (9) (2004) 1520–1533. doi:10.1109/TAC.2004.834113

[33] M. P. Allen, D. J. Tildesley, Computer Simulation of Liquids, Oxford University Press, 2017. doi:10.1093/oso/9780198803195.001.0001

[34] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Annals of Data Science 2 (2) (2015) 165–193. doi:10.1007/s40745-015-0040-1

[35] J. Gan, Y. Tao, Dbscan revisited: Mis-claim, un-fixability, and approximation, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, SIGMOD ’15, Association for Computing Machinery, New York, NY, USA, 2015, p. 519–530. doi:10.1145/2723372.2737792

[36] Clustering datasets
URL http://cs.joensuu.fi/sipu/datasets/

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[38] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. 8th Int’l Conf. Computer Vision, Vol. 2, 2001, pp. 416–423.

[39] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (6) (1986) 679–698. doi:10.1109/TPAMI.1986.4767851

[40] P. Erdős, J. Pach, R. Pollack, Z. Tuza, Radius, diameter, and minimum degree, J. Combin. Theory Ser. B 47 (1) (1989) 73–79. doi:10.1016/0095-8956(89)90066-X

[41] B. Mohar, Eigenvalues, diameter, and mean distance in graphs, Graphs Combin. 7 (1) (1991) 53–64. doi:10.1007/BF01789463