3D-SSF: A bio-inspired approach for dynamic multi-subject clustering of white matter tracts

A. Chekir, S. Hassas, M. Descoteaux, M. Côté, E. Garyfallidis, F. Oulebsir-Boumghar

PII: S0010-4825(17)30023-9
DOI: http://dx.doi.org/10.1016/j.compbiomed.2017.01.016
Reference: CBM2588

To appear in: Computers in Biology and Medicine

Received date: 30 September 2016
Revised date: 22 January 2017
Accepted date: 25 January 2017


This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting galley proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
3D-SSF: A Bio-inspired Approach for Dynamic Multi-Subject Clustering of White Matter Tracts

A. Chekir\(^a\)\(^*,\) S. Hassas\(^b\), M. Descoteaux\(^c\), M. Côté\(^c\), E. Garyfallidis\(^c\), F. Oulebsir-Boumghar\(^a\)

\(^a\)USTHB university, FEI, LRPE, ParIMéd, Algiers, Algeria
\(^b\)Université Lyon 1, LIRIS, UMR5205, F-69622, France
\(^c\)Sherbrooke Connectivity Imaging Lab, Computer Science, Université de Sherbrooke, Sherbrooke, QC, Canada

Abstract

There is growing interest in the study of white matter (WM) variation across subjects, and in particular the analysis of specific WM bundles, to better understand brain development and aging, as well as to improve early detection of some diseases. Several WM multi-subject clustering methods have been proposed to study WM bundles. These methods aim to overcome the complexity of the problem, which includes the huge size of the WM tractography datasets generated from multiple subjects, the existence of various streamlines with different positions, lengths and geometric forms, as well as the presence of outliers. However, the current methods are not sufficiently flexible to address all of these constraints. Here we introduce a novel dynamic multi-subject clustering framework based on a distributed multiagent implementation of the Multiple Species Flocking model, that we name 3D-Streamlines Stream Flocking (3D-SSF). Specifically, we consider streamlines from different subjects as data streams, and each streamline is assigned to a mobile agent. Agents work together following flocking rules in order to form a flock. Thanks to a similarity function, the agents that are associated with similar streamlines form a flock, whereas the agents that are associated with dissimilar streamlines are considered outliers. We use various experiments performed on noisy synthetic and real human brain data to validate 3D-SSF and demonstrate that it is more efficient and robust to outliers compared to other classical approaches. 3D-SSF is able to extract WM bundles at a population level, while considering WM variation across subjects and eliminating outlier streamlines.

Keywords: Data stream, multi-agent system, Multiple Species Flocking model, multi-subject clustering, outliers, White Matter tractography.

1 INTRODUCTION

Diffusion Magnetic Resonance Imaging (dMRI) \([1, 2]\) is a powerful, non-invasive imaging tool for the study of WM in vivo based on the measurement of water molecule diffusion in human brain tissue.

The diffusion is represented by a model \([3]\) that estimates the underlying fiber orientations based on the average displacement of water molecules for every voxel along several directions. The fiber orientations are exploited to trace 3D trajectories representing the potential pathway of WM via a process called tractography \([4]\). In this paper, we use the term ”streamline” to describe a set of 3D points representing a virtual anatomical fiber \([5]\).

The clustering of the streamlines into anatomically meaningful bundles is a very important task. Streamline clustering is used to study brain connectivity and contribute to a better understanding of the human brain. A whole-brain tractography dataset is highly complex, ranging from 100,000 to 1,000,000 streamlines per subject. Streamline clustering is challenging due to the following reasons:

- The complexity of WM architecture and frequently unclear definitions of the structure and the organization of WM bundles \([6]\).
- The huge size of the WM tractography dataset composed of various streamlines with complex geometries.
- The variability of human brain WM architecture across subjects in terms of geometric properties, connectivity strengths, and patterns, which complicates the development of a generalized solution for streamline clustering.
- The presence of outliers due to tractography errors and patient movements, which may distort the clustering results via erroneous streamlines.

Given these constraints, several streamlines clustering techniques have been proposed in the literature and can be divided into two main categories. The first uses regions of interest (ROIs) to select and/or exclude streamlines passing through these ROIs \([7]\). This strategy has been used to create WM bundle atlases \([8, 9]\) but requires extensive knowledge of brain anatomy and inten-
sive user interaction with a 3D environment. The second clustering technique category is based on a metric matrix computing the pairwise similarity between every streamline [10, 11, 12]. These methods are fully automatic but are limited by the number of streamlines they can analyze. Indeed, their complexity increases quadratically as the number of streamlines increases. For \( N \) streamlines, the size of the similarity matrix will be \( N \times N \) and require \( N \times (N - 1)/2 \) pairwise comparisons. For 100,000 streamlines, approximately 40 gigabytes (Gb) of space would be required for storage [13].

Some studies have attempted to reduce the computational complexity of automatic streamline clustering applied to a single subject. In 2012, Garyfallidis et al. proposed the QuickBundles (QB) algorithm as a means of reducing the size of tractography datasets, but the resulting clusters may not correspond to any anatomical bundle [14].

In 2015, Chekir et al. [15] proposed to use anatomical information provided by a bundles atlas to classify the resulting QB clusters. Several techniques based on hierarchical clustering using a divisive approach were also used to cope with the high number of streamlines [16, 17, 18, 19]. However, the results of these approaches depend heavily on the chosen number of hierarchical steps as well as several input parameters to obtain significant WM bundles [20].

The clustering of WM tractography datasets from a single subject is insufficient to establish a bundles model across a population because the WM architecture is variable from one subject to another and is very complex [6]. The streamlines from multiple subjects must be clustered together to create a model composed of a set of generic streamline bundles that can be detected in most of the population. Thus, multi-subject clustering presents an even greater challenge than WM clustering in a single subject.

In this paper, we are interested in multi-subject streamline clustering, which is essential for streamline-based population studies [20, 21, 13]. Streamlines from each subject cannot be clustered independently as determining which cluster from one subject corresponds to which cluster in another subject is not trivial. Ideally, the streamlines across all subjects would be clustered all together, but this becomes more challenging as the number of streamlines dramatically increases. One solution is to split the clustering process into two phases: learning and assignment. In the learning phase, we build, for each bundle, a model that can identify streamlines that belong to that bundle. Then, during the assignment phase, each streamline is assigned to one of the bundles found during the learning phase. These bundle models should be robust to WM variation across the population so that even streamlines from unseen subjects can be clustered.

Many approaches have been proposed that follow this learning/assignment principle. In [21], a Bayesian framework using a Hierarchical Dirichlet Process Mixture (HDPM) was used to learn the bundle models from training data without supervision, and this model was used as priors to cluster streamlines of new subjects. This method can cluster datasets of more than 120,000 streamlines, but larger datasets degrade the quality of the results. The authors in [13] proposed to cluster partitions of the data, repeatedly using hierarchical clustering, followed by analysis of the cluster assignments to create final clusters of all datasets in a scalable manner.

In [22], G. Prasad et al. used atlas-based labeling and maximum density paths to build the bundle models. More recently, L. Dodero et al. proposed a method based on a Dominant Sets algorithm using two levels of clustering [20].

The first level of clustering is intra-subject clustering, which to reduce the size of the tractography dataset by employing only homogeneous bundles. The second level is inter-subject clustering, which relies on a space-invariant metric to identify corresponding WM bundles across multiple subjects without using anatomical priors.

In [23], E. Garyfallidis et al. proposed Streamline-based Linear Registration (SLR) based on the QB algorithm [14], which can also be used for multi-subject clustering.

Multi-subject clustering approaches have also been applied to create atlases, which support WM variance among multiple subjects [24, 25, 26]. However, apart from a few recent methods such as those described in [20, 18, 27], most approaches do not incorporate outlier rejection, which is essential for a robust clustering result. Outlier rejection allows the elimination of erroneous streamlines produced by low-quality data and tractography errors. For instance, in [20], the authors used Gaussian priors and statistical tests to remove outliers from intra-subject clustering. In [18], they incorporated Local Outlier Factors (LOFs) in CASTER (Cluster Analysis Through Smartly Extracted Representatives), an agglomerative hierarchical clustering method, to capture a relative degree of streamline isolation. In [27], they proposed the Hierarchical QuickBundle (HQB) method, which uses a clustering tree built from multiple applications of the QB algorithm. However, the results of this method depend strongly on a manually defined outlier threshold.

None of the methods proposed in the literature satisfactorily addresses all four issues described in the beginning of this section. Because of their static character, the definite clustering of a streamline in a given class and learned bundle models to cluster streamlines of unseen subjects are not always straightforward. Consequently, these methods are limited dynamically in considering the data variation issued from several subjects.

In the last few years, many clustering approaches based on the bio-inspired paradigm have been proposed. The authors in [28] studied the Flocking model and proposed a Multiple Species Flocking model (MSF) applied to text clustering. Recently, two methods in [29, 30] combined the MSF model and the DenStream algorithm [31] based on density for data stream clustering. However, none of these approaches has been applied to the problem of streamline clustering, although [32] recently used the flocking model for brain tractography.
In this work, we propose a novel framework inspired by the MSF model that permits dynamic and automatic multi-subject streamline clustering. We named our technique 3D-Streams Stream Flocking (3D-SSF). Our method considers the WM streamlines from various subjects feeding the system during regular intervals as a sequential data stream. This streamline stream is divided into chunks. Each chunk is operated independently, and the history statistics representing past information issued from the clustering of the previous chunk are inserted in the next one. This approach makes our method scalable to the huge tractography datasets generated from all subjects.

The proposed framework uses a multi-agent system with distinct simple functionalities to mimic the flocking behavior and operates in two phases. In the initialization phase, we consider only the first chunk of the streamline stream. Each streamline is assigned to a SimpleAgent deployed in a 3D virtual space and working together with other agents, following flocking rules. Agents move in the space for a fixed time, and when they encounter other agents in their perception area (circle with a fixed radius), they evaluate their similarity measure and then decide to form a flock (if they are similar) or not. An agent can leave a flock for another containing agents with higher similarity during a move, resulting in dynamic clustering. Thus, only agents associated with similar streamlines form a flock. The outliers are the streamlines associated with isolated agents or streamlines associated with agents belonging to flocks with a small number of streamlines. At the end of this phase, history statistics are collected: for each flock, we estimate a representative streamline called the medoid, which engenders two other types of agents: PotentialAgents carrying potential medoids, representing potential flocks, and OutlierAgents carrying outlier medoids, representing outlier flocks.

The second phase is called summary statistics maintenance and clustering. In this phase, we operate on new data stream chunks in a sequential way. The new streamlines are assigned to SimpleAgents and attempt to merge with the previous PotentialAgents and OutlierAgents to form flocks of similar streamlines during a predefined number of iterations. As a result of this second phase, the history statistics are updated, and the final bundles are generated using these representative streamlines without outliers thanks to the new adaptive method used for outlier elimination.

We have evaluated our proposed method in different experiments on synthetic and real dMRI datasets. We have also performed a quantitative analysis comparing our results with other solutions in the literature [14, 33, 23]. Our results demonstrate the effectiveness and accuracy of our approach.

In the organization of this paper, section 2 details our proposed method and the associated concepts. Section 3 describes the pre-processing achieved on different data and the experiments and presents the obtained results. A comparison of our method to existing approaches is presented as well as a discussion of the obtained results. A general conclusion of the presented work and its results is presented in section 4.

2 METHODS

2.1 Fundamental Concepts

The method presented in this paper reassembles concepts of the MSF model [28], streamline similarity measure [34] and the adaptive outlier detection mechanism, which are detailed in the following section.

2.1.1 Streamline similarity measure

Different similarity measures have been proposed in the literature and can be divided into two categories: voxel-based similarity methods and streamline-based similarity methods. The voxel-based similarity methods compute the difference in the diffusion characteristics between voxels connected by the streamlines [35, 36]. This type of method has good scalability but requires fastidious post- and pre-processing steps. The second type of method gives more precise results but suffers from problems of scalability, except the QB method using the Minimum average Direct-Flip distance (MDF) [14]. Several strategies have been proposed to cluster streamlines based on the distance matrix. The first is obtained by computing the Euclidean distance between the mass center of two streamlines. The second one defines the similarity between streamlines, such as the distance between their extremities [37]. The drawback of this metric is the penalization of the streamlines with distant extremities belonging to the same bundle, for example, the corticospinal. The third strategy is a point to point comparison along the streamline, which requires that the compared streamlines have the same number of points, using different metrics such as the Closest Distance [38, 39], Mean Closest Distance [40] and Hausdorff distance [41].

In [34], the authors performed a comparative study between several similarity measures and demonstrated that the mean of closest distances $d_M$ is more suitable for a good representation of the average resemblance between streamlines, whereas the Hausdorff distance $d_H$ appears more accurate if a stringent similarity measure is required.

To achieve precise clustering results, in our framework, we have opted for the Hausdorff distance to estimate the similarity value between streamlines inspired from [34].

Let $S_i$ be a streamline represented by a set of 3D points $P_k$. The Hausdorff distance ($d_H$) is defined as the maximum of the closest distances for every point of streamline $S_i$ to streamline $S_j$:

$$d_H(S_i, S_j) = \max(d_h(S_i, S_j), d_h(S_j, S_i))$$ (1)

with $d_h(S_i, S_j) = \max_{P_k \in S_i, \min_{P_{l} \in S_j}} \|P_k - P_l\|$
2.1.2 Multiple Species Flocking Model

The MSF is a bio-inspired computational model for simulating the animation of a heterogeneous flock of entities developed in [28]. It represents movements in a group of entities, as seen in a flock of birds and a school of fish. In this model, each individual (called boid) makes movement decisions without any explicit communication with others. Instead, it acts according to four simple rules. Of them, the three basic rules were first proposed by Reynolds [42]: (1) Alignment: Steer towards the average direction of movement of nearby agents. (2) Separation: Steer to avoid being too close to nearby agents. (3) Cohesion: Steer to move toward the center position of nearby agents. A fourth rule, the similarity rule, has been added to the basic rules in [28], allowing for each boid to discriminate among its neighbors and to group only with those similar to itself.

The similarity rule enables the flock to organize groups of heterogeneous boids into homogeneous subgroups consisting only of boids of the same species by using the concept of velocity vector. The velocity vector for each agent is calculated by summing the velocity vectors given by the similarity, alignment, cohesion, and separation rules.

\[ \vec{V} = \vec{V}_{ar} + \vec{V}_{sr} + \vec{V}_{cr} + \vec{V}_{sim} + \vec{V}_{dsim} \]  (2)

Alignment rule

As illustrated in Figure 1, the boid tends to move in the same direction as that of the nearby boids, i.e., it gives an agent the capability to align its velocity vector with the average velocity vector of other nearby agents in its visibility range. It is described by the following formula:

\[ \text{if } (d(F_i, A_c) \leq \epsilon) \land (d(F_i, A_c) \geq \lambda) \Rightarrow v_{ar} = \frac{1}{n} \sum_{i} \vec{v}_i \]  (3)

where \( v_{ar} \) is the velocity driven by the alignment rule, \( \epsilon \) the radius indicating the visibility range of the boids, \( \lambda \) the minimum distance that must be maintained among them respectively (with \( \epsilon > \lambda \)), \( d(F_i, A_c) \) is the Euclidean distance between the agent \( A_c \) and its neighbor flockmate \( F_i \), \( \vec{v}_i \) is the velocity of the agent \( F_i \) and \( n \) is the number of neighbor agents.

Separation rule

As illustrated in Figure 2, it gives an agent the capacity to maintain an established distance from other boids nearby. It is described by the following formula:

\[ \text{if } d(F_i, A_c) \leq 2 \ast \lambda \Rightarrow v_{sr} = \sum_{i} \frac{v_i + v_c}{d(F_i, A_c)} \]  (4)

where \( v_{sr} \) is the separation velocity, \( v_c \) is the velocity of the current agent and \( v_i \) is the velocity of the \( i^{th} \) flock mate.

Cohesion rule

As illustrated in Figure 3, the agent’s velocity vector is aligned with the direction of the average spatial position of the agents within its visibility range. It is described by the following formula:

\[ \text{if } (d(F_i, A_c) \leq \epsilon) \land (d(F_i, A_c) \geq \lambda) \Rightarrow v_{cr} = \sum_{i} (P_i - P_c) \]  (5)

where \( v_{cr} \) is the cohesion velocity and \( P_i, P_c \) are the positions of the current agent.

Similarity rule

The similar agents use the strength of the attracting force to stay in the same flock. It is computed as:

\[ V_{sim} = \sum_{i} (Sim(F_i, A_c) \ast d(P_i, P_c)) \]  (6)

Figure 1: Alignment Rule. Agent steers toward the average heading of local neighborhood agents in its visibility range \( \epsilon \).

Figure 2: Separation Rule. Agent steers to avoid crowding neighborhood agents in its visibility range by maintaining among them the minimum distance \( \lambda \).

Figure 3: Cohesion Rule. Agent steers to move toward the average position of neighborhood agents.
The dissimilar agents use a repulsive force to stay away from each other. It is computed as:

\[ V_{dsim} = \sum_{i=1}^{N} \frac{1}{\text{Sim}(F_i, A_c) \times d(P_i, P_c)} \]  

(7)

where \( V_{sim}, V_{dsim} \) are the similarity velocity and the dissimilarity velocity respectively, \( \text{Sim}(F_i, A_c) \) is the Hausdorff distance in our case (detailed in section 2.1.1), used to estimate the feature of streamlines carried by agents \( F_i \) and \( A_c \), and \( d(P_i, P_c) \) is the Euclidean distance between their positions.

2.1.3 Adaptive mechanism for outlier detection

We propose in this work a new approach for outlier detection based on two verification levels using adaptive thresholds. A cluster is considered an outlier cluster if it satisfies one of these two conditions:

- The first verification level is based on the number of streamlines that constitute every cluster. The entire cluster is selected as an outlier cluster if it contains less than the average number of objects \( \text{SizeThreshold} = (n/k) \), where \( n \) is the number of streamlines of the current chunk and \( k \) is the number of clusters obtained with our clustering method on the current chunk.

- In the second verification level, for each cluster, the absolute distance between its medoid and the medoids of other clusters is calculated. The medoid streamline of a flock is defined as that streamline for which the average dissimilarity to all other streamlines in the flock is minimal. This distance is called the Absolute Distance between the Medoid and Medoids (ADMM). For each cluster, the isolation degree is the lowest possible input value of similarity required to join two streamlines in one cluster. The value of the isolation degree is calculated as the average of all ADMM values of all clusters in the current chunk multiplied by a constant \( Cst \) according to the following equation:

\[ \text{IsolationDegree} = (\text{Average(ADMM)}) \times Cst \]  

(8)

Henceforth, for each cluster, the ADMM value is calculated and compared with the corresponding IsolationDegree value. If the ADMM of a specific cluster is greater than its corresponding threshold value, it is considered an outlier.

2.2 3D-SSF Algorithm

The 3D-SSF algorithm adapts the above-described concepts for a dynamic multi-subject WM clustering framework allowing outlier elimination. The streamlines from various subjects in sequential order (one subject after another) are considered a stream of streamlines. This stream is divided into \( N \) disjoint chunks of equal size, and the 3D-SSF algorithm is applied to every chunk according to two distinct phases: the initialization phase and the summary statistics maintenance and clustering phase.

Our algorithm uses a multi-agent system to mimic multiplexerspecies flocking behavior. Each streamline is assigned to an agent determined by his position \( P = (x, y, z) \) and his velocity vector \( \vec{V} \). The agents are deployed randomly in 3D space.

Through the different phases, three types of agents are introduced: SimpleAgent, PotentialAgent and OutlierAgent.

Initialization phase

At the beginning, when the first chunk of the stream is introduced into the system, every streamline is associated with an agent called the SimpleAgent. The SimpleAgents work in parallel for a predefined number of iterations to group agents carrying similar streamlines together, exploiting the similarity function (detailed in section 2.1.1).

Each agent compares his similarity function with only his neighbor’s agents. The agent is considered a neighbor of another agent if it is in its visibility radius, i.e., the direct distance (DD) (equation 9) between the two agent positions, defined as the sum of the absolute differences of their Cartesian coordinates, is below a fixed threshold \( \epsilon \).

\[ \text{DD}(A_1, A_2) = |x_2 - x_1| + |y_2 - y_1| + |z_2 - z_1| \]  

(9)

Where \( (x_1, y_1, z_1) \) and \( (x_2, y_2, z_2) \) are the positions of \( A_1 \) and \( A_2 \), respectively, in 3D space.

The agent’s velocity is calculated by applying the local rules of Reynolds with the similarity and dissimilarity rules following equation 2. In particular, the velocity vector for each agent is calculated by summing the velocity vectors given by the similarity, dissimilarity, alignment, cohesion and separation rules to form a flock composed only of similar agents. Based on the similarity and dissimilarity rules, the flocking agent attempts to remain close to the agents with similar features and away from other agents with dissimilar features.

Consequently, the similarity rule confers an adaptive behavior of the agent, and the participation of an agent in a group is not definitive. If a more similar agent is encountered during the space exploration, the current flock can be dropped, and the agent joins another flock with higher similarity.

Several flocks (swarms) are obtained at the end of this phase. These swarms are divided following the adaptive mechanism for outlier detection detailed previously (section 2.1.3) into two types: potential swarms and outlier swarms. The potential swarm is represented by the potential medoid, assigned to a new introduced agent called the PotentialAgent. The outlier swarm is represented by an outlier medoid, assigned to a new introduced agent called the OutlierAgent.
Summary statistics maintenance and clustering phase

When new chunks of the stream are introduced into the system one by one, the streamlines constituting it are assigned to SimpleAgents and attempt to merge with the potential and outlier agents result from the previous steps. Different cases can occur:

- A PotentialAgent \((A_{p1})/\text{OutlierAgent } (A_{o1})\) meets a PotentialAgent \((A_{p2})/\text{OutlierAgent } (A_{o2})\) in its perception range \((d(A_{p1}, A_{p2}) \leq \epsilon)\), the similarity between them is computed. If the agents are similar \((\text{equation } 1 < \text{Threshold Similarity } (\gamma))\), they then calculate the velocity vector by applying the Reynolds with similarity and dissimilarity rules \((\text{equation } 2)\) and join to form a swarm (i.e. cluster) of similar PotentialAgents/OutlierAgents.

- A SimpleAgent \((A_s)\) meets either a PotentialAgent \((A_{p})/\text{OutlierAgent } (A_{o})\) in its perception range. The similarity between them is calculated, and if they are similar, \((A_s)\) is absorbed by \((A_{p})/A_{o}\).

- A SimpleAgent \((A_s)\) meets another SimpleAgent \((A_{s2})\). The similarity between the two agents is calculated, and if the agents are similar, the velocity vector is computed, and \((A_s)\) joins \((A_{s2})\) to form a swarm of OutlierAgents.

During this movement, the agent can change its belonging to a swarm if it encounters a more similar swarm.

The summary statistics are updated at the end of a fixed maximum number of iterations by estimating the potential and outlier representatives for the newly obtained swarms. If the size of a PotentialAgent swarm decreases below a defined SizeThreshold or its ADMM value increases above a defined IsolationDegree, it is degraded to become an OutlierAgent swarm.

Otherwise, if the size of an OutlierAgent swarm increases above the SizeThreshold or its ADMM decreases below the IsolationDegree, a new PotentialAgent swarm is created.

Our method yields swarms of potential medoid streamlines at the end of the iteration, and outlier swarms are rejected.

The concept of the 3D-SSF algorithm is introduced by the pseudo code detailed in the appendix, and Figure 4 shows consecutive snapshots of how the process evolves for one bundle.

3 RESULTS AND DISCUSSIONS

The following section presents different experiments on real and synthetic datasets to evaluate the efficiency and accuracy of our method. The 3D-SSF algorithm was implemented in Python language. The running has been performed on an Intel (R) Core (TM) i7 with 8 GB of memory.

3.1 Experiments on Synthetic Data

3.1.1 Materials

To evaluate the capacity of our method to extract correctly various bundles and its robustness to outliers, it was applied to synthetic WM streamlines created using the Numerical Fibers Generator software [43], provided for HARDI reconstruction ISBI 2013 Challenge. This synthetic data has a spherical volume with a fixed radius and diameters ranging from 1.83mm to 7.27mm. It is composed by a total of 940 synthetic geometries divided into 20 simulated known ground truth WM bundles, imitating kissing, bending, branching, and crossing structures at angles between 30° and 90°, as seen in Figure 5(a). To evaluate the capacity of our method to eliminate the outliers, two other bundles composed by 4 lines and 3 curves were added to the synthetic data (dark red and purple bundles in Figure 5(a)), creating a total of 947 synthetic geometries.

3.1.2 Experiments

We implemented two other WM clustering algorithms: HQB clustering [27] and a V-Bundles method [33]. The HQB method is an improved version of the QB algorithm [14] that uses a probabilistic clustering tree from multiple applications of QB to allow outlier removal.

We performed the HQB clustering using MDF distance with \(s\) mm distance threshold. MDF distance requires streamlines to have the same number of points, the streamlines have been automatically downsampled to have only 20 points.

The V-Bundles method is based on an adaptation of the vector field k-means algorithm to extract WM bundles. We maintained the same parameters adopted in [33], the maximum number of k-means iterations= 3, \(\lambda = 0.001\), \(\gamma = 0.6\), \(r_v = 15mm\) and \(r_z = 9mm\).

Initially, we tested only the initialization phase with the entire synthetic data to evaluate the capacity of dynamic clustering based on the flocking model to reproduce effectively the synthetic data bundles. We employed a number of the initial agents corresponding to the total number of streamlines in the data \(N_o = 947\), and we determined threshold similarity \(\gamma = 8mm\) and max iteration number \(\text{MaxIterations } = 100\) as the optimal parameters and outlier parameters using the concept detailed in 2.1.3 (the constant of isolation degree was fixed at a value of 1.5).

We then executed all steps of our method (initialization and summary statistics maintenance and clustering phase) using the same parameters specified above except sample size because we divided the data into chunks of 100 streamlines. We then considered \(N_o = 100\).

3.1.3 Results

The results obtained for the synthetic data using the three methods (3D-SSF (Initial phase), HQB, V-Bundles) are shown in Figure 5. The dynamic WM clustering based
on flocking model was able to separate the streamlines into 22 meaningful clusters: 20 principal bundles represented by 20 potential medoid streamlines (the medoid is the black streamlines in bundles in Figure 5(b)) assigned to the potential agents and two outlier clusters represented by two outlier medoid streamlines carried by outlier agents that were eliminated from the final result. HQB successfully eliminated outliers but mixed two principal bundles, the yellow green and orange bundles in (Figure 5(a)), in the same cluster (orange bundle in Figure 5(c)). Thus, we obtained a total of 21 clusters with HQB: two outlier clusters and 19 principal bundles (see Figure 5(c))). However, the V-bundle method was disturbed by outliers; it extracted the 20 principal bundles, but they were mixed with outliers (Figure 5(d)).

The results obtained with the 3D-SSF method (with the two phases) demonstrate that our method can reproduce perfectly the ground truth of synthetic bundles by extracting the 22 clusters, 20 principal bundles and two outlier clusters. Each bundle is characterized by a set of medoid streamlines.

We performed a statistically robust comparison using the Adjusted Rand Index and the Completeness and Homogeneity indexes [44] to evaluate the accuracy and performance of the methods. Higher Completeness indicates that streamlines belonging to the same anatomical bundle are clustered together. The Adjusted Rand Index is defined as the number of agreement pairs divided by the total number of pairs. If the two bundles agree completely, then the Rand Index returns a value of 1; otherwise, the lower limit of this index is 0. The Homogeneity Index calculates the degree to which a cluster contains only streamlines of a single class.

Table 1 presents the results of this quantitative analysis. 3D-SSF showed higher Completeness, Adjusted Rand Index and Homogeneity, coinciding with the results obtained, which generated the same clusters of the ground truth. This experiment demonstrated the capacity of our method to eliminate outliers, which is ideal in our case because the real data are very noisy. The results of HQB also showed high Completeness but lower Homogeneity, indicating that the clusters were complete but did not contain streamlines from the same structure. V-bundles performed poorly with respect to both Homogeneity and Completeness compared to the other approaches, and thus the resulting clusters were typically not well separated.

<table>
<thead>
<tr>
<th>Method</th>
<th>Adjusted Rand Index</th>
<th>Completeness</th>
<th>Homogeneity</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D-SSF</td>
<td>0.955</td>
<td>0.981</td>
<td>0.981</td>
</tr>
<tr>
<td>HQB</td>
<td>0.838</td>
<td>0.920</td>
<td>0.899</td>
</tr>
<tr>
<td>V-Bundles</td>
<td>0.791</td>
<td>0.886</td>
<td>0.816</td>
</tr>
</tbody>
</table>

3.2 Experiments on Real Data
3.2.1 Materials

The proposed method was tested on dMRI real data of 10 healthy subjects (100307, 124422, 161713, 199655, 201111, 239944, 245333, 366446, 525446 and 856766), taken from the Human Connectome Project (HCP) [45, 46]. Data were acquired on a Siemens Skyra 3T scanner parameterized at 90 diffusion encoding gradients with a b-value of 1000s/mm² and isotropic spatial resolution of 2mm.

Local streamline orientations were estimated using CSD [47] based on Spherical Deconvolution (SD), which provides an estimate of the fiber orientation distribution (FOD). The response function was estimated using voxels with fractional anisotropy (FA) value greater than 0.7 and a maximum harmonic order of 8 was used.

A deterministic fiber tracking is performed by EuDX algorithm based on Euler integration and trilinear interpolation [48], with a threshold of 0.2, tracking stops in regions with FA below that threshold, to ensure to be in WM area. Tractography was initiated from 1,000,000 random seeds, then the size of tractography for one subject is about 1,000,000. The reconstruction and tractography were generated by DIPY software[49].
The HCP data of all subjects are in the standard space MNI 152, thus the clustering across a population can be done without any inter-subject registration.

3.2.2 Experiments

The 3D-SSF algorithm was tested on real data. The settings of our algorithm were chosen to obtain the best clustering results. We considered the data of the ten subjects as sequential flux and subdivided it into chunks of 10,000 streamlines. The number of initial agents \( N_0 = 10,000 \) (the same results are obtained with different chunk sizes due to the dynamic nature of our approach), threshold similarity \( \gamma = 8 \text{mm} \), the max iteration number \( \text{MaxIterations} = 100 \) and outlier parameters were determined using the concept detailed in 2.1.3.

We compared the clustering accuracy and time efficiency of our method with alternative techniques in the literature, such as the most recent approach, Multi-subject Clustering with QuickBundles (MSQB), proposed in [23], which is based on the QB algorithm and considered the fastest method in the literature [14].

We implemented this method following the same steps as in [23]. First, we removed streamlines with a length shorter than 50\text{mm}, and then we grouped the streamlines into bundles using the QB method with a 20\text{mm} distance threshold and downsampled the streamline to only 20 points. Each cluster was represented by a virtual streamline called a medoid. We repeated this process for each subject. Finally, we clustered the streamline medoid outcomes for each subject to obtain bundles representing the WM variation across the subject population. Both methods were compared to the bundle atlas constructed in this study.

3.2.2.1 Clustering accuracy

The main problem with WM clustering is the validation of any method on real datasets due to the lack of a ground truth. To evaluate the accuracy of our results on the real dataset described above, we constructed a bundle atlas as our gold standard.

Atlas Bundle construction We used white matter query language (WMQL)\(^1\) [50] to automatically segment the tractography datasets of our ten subjects according to a parcellation in FreeSurfer and definitions of white matter bundles. We have obtained for each subject 37 neuroanatomical bundles; several projection, association and commissural bundles, namely, the left/right corticospinal tract (CST), the left/right cingulum (CG), the left/right arcuate fasciculus (AF), the left/right inferior longitudinal fasciculus (ILF), the left/right inferior fronto-occipital fasciculus (IFO), the 1, 2, and 3 left/right superior longitudinal fasciculus (SLF 1, 2, 3); and finally, the corpus callosum (CC) in its seven parts (CC 1-7).

Afterward, we generated a bundle atlas of all these bundles using the method described in [51]. The bundles obtained with WMQL segmentation of each subject were converted to binary maps (1: pixels that contained the streamline, 0: pixels that did not contain the streamline). The binary maps obtained from the 10 subjects were averaged to produce a probabilistic WM map in the MNI standard.

The probabilistic maps of the bundles in the atlas were superimposed on the binary maps of the bundles obtained with our method and MSQB clustering. The spatial matching between the maps was examined for twelve bundles chosen randomly for statistical analysis of reproducibility as reported in [52]. We estimated the spatial matching ratio (SMR), spatial volume agreement (SVA) and kappa statistics between the atlas bundles, our bundle results and MSQB clustering bundles for comparison.

The results of the kappa analysis are illustrated in Figure 6(a). The Kappa values were quite high for all bundles obtained with our method. An average Kappa value of more than 0.80 was observed, versus 0.7 for MSQB clustering. A value of 0.11-0.2 is considered slight, 0.21-0.4 fair, 0.41-0.60 moderate, 0.61-0.80 substantial, and 0.81-1.0 almost perfect agreement.

The results of this experiment demonstrate the precision of our method. The left inferior longitudinal fasciculus (ILF-L) had the smallest index of agreement (Kappa=0.73), and the right corticospinal (CST-R) had the best agreement (Kappa = 0.93).

The average results of SMR and SVA for both methods are shown in Figure 6(b,c). The results of our method indicate a high level of matching for all bundles. The average SMR and SVA were 86% and 94% , respectively, for our method, versus 78% and 84% for MSQB clustering, demonstrating the good performance of the 3D-SSF approach compared with the MSQB method.

We also performed a quantitative diffusivity analysis by measuring the mean FA values of every bundle using the Trackvis tool. For each bundle, the FA value was calculated by averaging the FA values of voxels belonging to this bundle. The bundles obtained with our method and MSQB were superimposed on the FA probabilistic map constructed from the FA map of the ten subjects. In addition, the bundles obtained by WMQL segmentation were superimposed on their corresponding FA map, and the FA value of the bundle was the average of the FA values collected from the different subjects.

Figure 7 shows that the difference between the bundles of the atlas and our method was quite small.

For the purpose of quantification, the scale metrics difference (SMD) is defined as:

\[
D(B_A, B_m) = \frac{|S_A - S_m|}{S_m}
\]

where \( S_A \) is the FA average value obtained by our method or MSQB method and \( S_m \) is the value for the atlas bundles. Table 2 shows the SMD results. The average SMD

\(^1\)http://tract-querier.readthedocs.org/en/latest/
Figure 6: The spatial matching comparison between bundles obtained with 3D-SSF method. (a) Kappa analysis. (b) The spatial matching ratio (SMR) and (c) spatial volume agreement (SVA).

was quite small, indicating that the bundles obtained with our method were accurate compared with the atlas bundles.

These experimental results prove that 3D-SSF achieves high clustering quality comparable to that of MSQB and affirm the ability of our method to properly extract the principal WM bundles. Figure 8 illustrates some of the major bundles obtained with our method such as the left/right corticospinal tract (CST), the left/right cingulum (CG), the left/right inferior longitudinal fasciculus (ILF), the left/right inferior fronto-occipital fasciculus (IFO), the left/right arcuate fasciculus (AF) and finally, the seven-part of corpus callosum (CC).

3.2.2.2 Computational efficiency

To assess the computational efficiency of our clustering method, runtime analysis was performed by examining the computing time for each clustering chunk on one subject chosen randomly and the computing time for clustering the tractography dataset composed of the ten HCP subjects. We used the fastest method in the literature, MSQB (detailed above), for the comparative study on the multi-subject clustering.

The run-time complexity of our method is of linear order because the agent must compare its characteristics with only neighborhood agents. For $N$ streamlines, the complexity is $O(NkT)$, where $k$ is the number of surrounding streamlines in the visibility range of each agent and $T$ is the number of iterations ($T = 100$ in our case). By contrast, most methods in the literature have a complexity of $O(N^2)$ because they require the comparison of all $N$ streamlines in the tractography dataset between them. These methods require much more time than our method to cluster any dataset; consequently, we chose the QB method for comparison on one subject. The QB method is
of complexity $O(N)$ in the best case, where $N$ is the total number of streamlines.

Figure 9 shows the time required to cluster a real dataset from one subject containing approximately 800,000 streamlines in steps of 10,000 (the size of a chunk). The results of this experiment show that our method requires a short computing time to finish the initial chunks. However, the computing time subsequently increases for other chunks because the chunks grow after injection of the representative streamlines issued from the clustering of the previous chunk.

Figure 10 shows the time required by our method to cluster the dataset composed of the ten HCP subjects compared with the MSQB method. Our method requires 1027 seconds to acquire good clustering results for one subject dataset of about 800,000 streamlines, whereas QB requires 370.3 seconds. The time required for every subject depends on the number of streamline components in it. To accomplish the clustering of the ten subjects, our method requires 10,701 seconds (approximately 3 hours), whereas MSQB requires 5,257 seconds (approximately one hour and a half) to cluster streamlines in the same dataset.

The results of this experiment demonstrate that our method exhibits acceptable time performance with respect to the huge size of the dataset and compared with the fastest method in the literature. Our method requires slightly more time to achieve a more precise clustering result than the MSQB method, as demonstrated by experiment 3.2.2.1, especially considering that the application is not in real time, which makes our method scalable for a very large streamline dataset.

3.2.2.3 Effectiveness of the velocity vector calculation

In this experiment, we demonstrate the effectiveness of the velocity vector calculation based on the combination of the four rules and explore some of its characteristics. 3D-SSF clustering was applied on the real dataset with different agent velocity vector calculation methods. In the first case, the velocity vector for each agent was calculated by summing the velocity vectors given by the similarity, alignment, cohesion and separation rules following equation 2. In the second case, we applied only Reynolds rules (alignment, cohesion and separation) without the similarity rule. Finally, in the third case, the velocity vector for each agent was calculated by summing the velocity vectors given by the alignment, cohesion, separation and similarity rules, but the similarity velocity was defined as the difference between the similarity velocity $V_{\text{sim}}$ (equation 6) and the dissimilarity velocity $V_{\text{dissim}}$ (equation 7).

Furthermore, to investigate the performance of the Hausdorff distance used to estimate the similarity between the streamline features carried by the agents $((\text{Sim}(F_i; A_e)$ in the equations 6 and 7), 3D-SSF clustering was performed twice on the real dataset using the same method of velocity vector calculation each time but with different streamline similarity measures. The Hausdorff distance (HD) and Minimum average Distance Flipped (MDF) proposed in [14] were used.

The bundles resulting from these different cases were
compared to our gold standard (the WM atlas constructed in this study) using the clustering validation index as the Adjusted Rand Index described above to evaluate the performance of the 3D-SSF clustering with different velocity vector calculation methods. Table 3 presents the results of this quantitative evaluation.

The results obtained using the Hausdorff distance are generally better for the three cases, which coincides with the study in [34]. We confirmed that the Hausdorff distance is more accurate than the MDF distance for the streamline similarity measure.

The first and third cases give the best results compared with the second case for all bundles. The addition of the similarity rule clearly improves the precision of the clustering results and the rapid formation of flocks composed only of similar streamlines. This performance can be explained by the capacity of the similarity velocity to maintain similar agents in the same flock and the capacity of the dissimilarity velocity to push away the other agents carrying dissimilar streamlines. By contrast, in the second case, in which only the three Reynolds rules were applied, the 3D-SSF clustering method was limited to correctly separate the streamlines into coherent bundles confused by outliers. The third case achieves results nearly identical to those of the first case.

4 Conclusion

In this paper, we presented a 3D-SSF algorithm allowing the identification of common bundles across subjects for the characterization of WM structure in a population. The proposed solution uses a multi-agent system to mimic the MSF model and thus is dynamic to address WM variability and very scalable to huge datasets generated from many subjects, thanks to the multi-agent paradigm features such as adaptivity, parallelism, asynchronism, and decentralization.

Our framework is a complete, precise solution that is able to discriminate bundles and completely eliminate outliers. The performance of our method was confirmed in various experiments using synthetic and real dMRI datasets. The experiments revealed good results compared with existing techniques. The main contributions of this work compared with others methods are as follows:

- The capacity to cluster dynamically a multi-subject data stream, thus allowing support of WM variability across subjects regardless of subject processing order due to the heuristic searching mechanism. This mechanism helps agents to quickly form a flock and re-generate new results rapidly: during a running process, agents can leave the group for another group containing agents with higher similarity.
- Scalability to very large data sets due to the division of data into chunks, which retains only a data chunk in memory and allows the computation of summary statistics used to cluster the next chunk.
- Clustering results precision, achieved through the consideration of past information to group the new streamline chunk via the summarization principle.
- Robustness to noise and outliers through the proposed novel concept for the definition of adaptive outlier thresholds. The experimental results show that our method obtains clusters of high quality even when outliers are present.

However, our framework uses only one metric (the Hausdorff distance) to measure the similarity between streamlines, whereas streamlines have complex geometric forms, with different lengths, shapes, orientations and spatial positions. Therefore, we plan in our future work to extend this approach by incorporating various similarity features (multiple similarity metrics) with different thresholds to reinforce the decision criteria and increase accuracy, exploiting agent autonomy, and decentralize the system.

We also plan to apply this approach to study the short association streamlines of superficial white matter, which have rarely been studied because of the limitations of most clustering methods with respect to tractography dataset size and its considerable variation. The code for 3D-SSF is freely available at https://github.com/amiraCHEK/3D-SSF-work-in-progress

References


The pseudo-code of the 3D-SSF algorithm

In this section we present the detailed pseudo-code of the 3D-SSF algorithm to more explain the concept of the proposed method, we begin by introducing the procedure allowing the summary statistics computing, afterward, we describe the 3D-SSF algorithm steps, including the initialization phase and the summary statistics maintenance and clustering phase.
Figure 4: Example of the 3D-SSF algorithm applied to the corticospinal tract left (CST-L) across different subjects. (a) The potential swarms resulting from different clustering steps with different colors of one subject. (b) The summary statistics representing potential swarms (potential medoids) affected to PotentialAgent. (c) The outlier swarms resulting from different clustering steps with different colors of one subject. (d) The summary statistics representing outlier swarms (outlier medoids) affected to OutlierAgent. (E) The final swarm results representing the CST-L bundle without outlier. (F) The final outlier swarm result.

Table 2: The SMD between FA values of bundle atlas and bundles obtained with our method and MSQB bundles.

<table>
<thead>
<tr>
<th></th>
<th>CSTL</th>
<th>CSTR</th>
<th>CGL</th>
<th>CGR</th>
<th>IFOL</th>
<th>IFOR</th>
<th>ILFL</th>
<th>ILFR</th>
<th>ORL</th>
<th>ORR</th>
<th>SLFL</th>
<th>SLFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D-SSF</td>
<td>0.142</td>
<td>0.121</td>
<td>0.129</td>
<td>0.142</td>
<td>0.285</td>
<td>0.341</td>
<td>0.028</td>
<td>0.052</td>
<td>0.147</td>
<td>0.05</td>
<td>0</td>
<td>0.22</td>
</tr>
<tr>
<td>MSQB</td>
<td>0.981</td>
<td>0.920</td>
<td>0.886</td>
<td>0.955</td>
<td>0.838</td>
<td>0.791</td>
<td>0.142</td>
<td>0.078</td>
<td>0.096</td>
<td>0.355</td>
<td>0.135</td>
<td>0.088</td>
</tr>
</tbody>
</table>

Table 3: The Adjusted Rand Index calculated between bundles obtained with 3D-SSF algorithm for different cases and atlas bundles.

<table>
<thead>
<tr>
<th></th>
<th>CSTL</th>
<th>CSTR</th>
<th>CGL</th>
<th>CGR</th>
<th>IFOL</th>
<th>IFOR</th>
<th>ILFL</th>
<th>ILFR</th>
<th>ORL</th>
<th>ORR</th>
<th>SLFL</th>
<th>SLFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1 (HD)</td>
<td>0.986</td>
<td>0.992</td>
<td>0.982</td>
<td>0.892</td>
<td>0.777</td>
<td>0.772</td>
<td>0.689</td>
<td>0.699</td>
<td>0.800</td>
<td>0.885</td>
<td>0.752</td>
<td>0.771</td>
</tr>
<tr>
<td>Case 1 (MDF)</td>
<td>0.899</td>
<td>0.923</td>
<td>0.844</td>
<td>0.782</td>
<td>0.728</td>
<td>0.713</td>
<td>0.699</td>
<td>0.699</td>
<td>0.800</td>
<td>0.885</td>
<td>0.752</td>
<td>0.771</td>
</tr>
<tr>
<td>Case 2 (HD)</td>
<td>0.561</td>
<td>0.586</td>
<td>0.450</td>
<td>0.488</td>
<td>0.299</td>
<td>0.332</td>
<td>0.388</td>
<td>0.395</td>
<td>0.411</td>
<td>0.513</td>
<td>0.355</td>
<td>0.447</td>
</tr>
<tr>
<td>Case 2 (MDF)</td>
<td>0.441</td>
<td>0.458</td>
<td>0.282</td>
<td>0.290</td>
<td>0.235</td>
<td>0.199</td>
<td>0.259</td>
<td>0.210</td>
<td>0.301</td>
<td>0.422</td>
<td>0.315</td>
<td>0.335</td>
</tr>
<tr>
<td>Case 3 (HD)</td>
<td>0.988</td>
<td>0.981</td>
<td>0.957</td>
<td>0.947</td>
<td>0.819</td>
<td>0.830</td>
<td>0.885</td>
<td>0.873</td>
<td>0.966</td>
<td>0.962</td>
<td>0.812</td>
<td>0.855</td>
</tr>
<tr>
<td>Case 3 (MDF)</td>
<td>0.948</td>
<td>0.955</td>
<td>0.899</td>
<td>0.821</td>
<td>0.730</td>
<td>0.754</td>
<td>0.738</td>
<td>0.785</td>
<td>0.857</td>
<td>0.911</td>
<td>0.795</td>
<td>0.720</td>
</tr>
</tbody>
</table>
Algorithm 1 Compute Summary Statistics Function

1: procedure Summary Statistics(swarms)
2:     for Each Swarm do
3:         compute medoid
4:     for Each medoid M do
5:         ADMM = EuclideanDistance(medoid M, medoid)
6:         IsolationDegree = (average(ADMM) * Cst)
7:     end for
8:     if (len(swarm) > (len(ChunkStream)/NumberOfSwarms)) and (ADMM < IsolationDegree) then
9:         PotentialMedoid = medoid
10:         Append (PotentialMedoid) to PotentialSwarm
11:     else if then
12:         OutlierMedoid = medoid
13:         Append (OutlierMedoid) to OutlierSwarm
14:     end if
15: end for
16: return PotentialMedoid, OutlierMedoid
17: end procedure

Algorithm 2 3D-SSF Algorithm- Initialization phase

Input: Streamlines of the first stream chunk
Output: PotentialSwarms, OutlierSwarms

for i = 1 to MaxIterations do
2:     for streamline in ChunkStream do
3:         SimpleAgent = streamline
4:     end for
5:     for Each SimpleAgent X do
6:         if DirectDistance(X, Y) < ε (X contains in its visibility range a SimpleAgent Y) then
7:             if similarity(X, Y) < γ then
8:                 computeVelocity(flock, MSF rules) according to equation2
9:                 Move SimpleAgent: PositionSimpleAgent = PositionSimpleAgent + \vec{V}_{velocity}
10:             end if
11:         end if
12:     end for
13: end for
14: Compute Summary Statistics Function(swarms)
15:     Affect PotentialMedoid to PotentialAgent and OutlierMedoid to OutlierAgent
16: end for
Algorithm 3 3D-SSF Algorithm: Summary statistics maintenance and clustering phase

**Input:** Streamlines of the chunk stream, PotentialMedoids, OutlierMedoids

**Output:** PotentialSwarms, OutlierSwarms

Repeat for every chunk in stream:

1. for \( i = 1 \) to \( MaxIterations \) do
   2. for streamline in NewChunkStream do
      3. SimpleAgent = streamline
         end for
   4. for Each PotentialAgent X do
      5. if DirectDistance\((X, Y) < \epsilon \) (X contains in its visibility range a PotentialAgent Y) then
         6. if similarity\((X, Y) < \gamma \) then
            7. computeVelocityVector(flock, MSF rules)
            8. Move PotentialAgent: \( Position_{PotentialAgent} = Position_{PotentialAgent} + \vec{V}_{velocity} \)
            9. Append (PotentialAgent) to PotentialSwarm
         end if
      end if
   end for
   10. for Each OutlierAgent X do
      11. if DirectDistance\((X, Y) < \epsilon \) (X contains in its visibility range a OutlierAgent Y) then
         12. if similarity\((X, Y) < \gamma \) then
            13. computeVelocityVector(flock, MSF rules)
            14. Move OutlierAgent: \( Position_{OutlierAgent} = Position_{OutlierAgent} + \vec{V}_{velocity} \)
            15. Append (OutlierAgent) to OutlierSwarm
         end if
      end if
   end for
   16. for Each SimpleAgent do
      17. if DirectDistance\((X, Y) < \epsilon \) (X contains in its visibility range a OutlierAgent Y) then
         18. SimpleAgent X is temporary absorbed in the OutlierSwarm
         19. else if DirectDistance\((X, Y) < \epsilon \) (X contains in its visibility range a SimpleAgent Y) then
            20. if similarity\((X, Y) < \gamma \) then
               21. computeVelocityVector(flock, MSF rules)
               22. Move SimpleAgent: \( Position_{SimpleAgent} = Position_{SimpleAgent} + \vec{V}_{velocity} \)
               23. Append (SimpleAgent) to OutlierSwarm
            end if
         end if
      end if
   end for
   24. Compute Summary Statistics Function (PotentialSwarms)
   25. Compute Summary Statistics Function (OutlierSwarms)