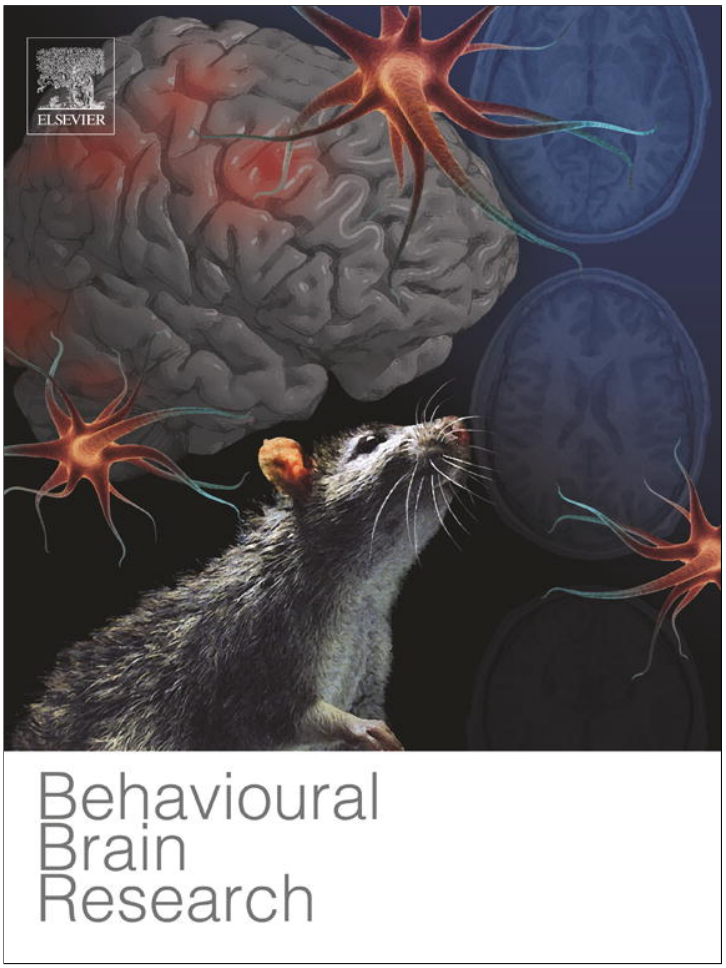

(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Behavioural Brain Research 241 (2013) 38–49

Contents lists available at SciVerse ScienceDirect

Behavioural Brain Research

journal homepage: www.elsevier.com/locate/bbr

Research report

Determining shoal membership using affinity propagation

Vicenç Quera [a,*], Francesc S. Beltran [a], Inmar E. Givoni [b], Ruth Dolado [a]

[a] Institute for Brain, Cognition and Behavior (IR3C), Adaptive Behavior and Interaction Research Group (GCAI), Department of Behavioral Science Methods, University of Barcelona, Campus Mundet, Passeig Vall d'Hebron 171, 08035 Barcelona, Spain
[b] Department of Computer Science, University of Toronto, 10 King's College Road, Toronto, Ontario M5S 3G4, Canada

HIGHLIGHTS

- We propose using the affinity propagation clustering for detecting multiple shoals.
- A soft temporal constraint is included in order to detect shoal fusion and fission.
- We explore how affinity propagation performs on agent-based simulated shoals.
- We compare affinity propagation clustering to human clustering of the same data.
- Affinity propagation is an appealing approach for detecting shoal dynamics.

ARTICLE INFO

Article history:
Received 31 August 2012
Received in revised form
16 November 2012
Accepted 20 November 2012
Available online xxx

Keywords:
Animal group membership
Shoal membership
Shoal fusion and fission
Affinity propagation clustering
Soft temporal constraint
Human clustering validation

ABSTRACT

We propose using the affinity propagation (AP) clustering algorithm for detecting multiple disjoint shoals, and we present an extension of AP, denoted by STAP, that can be applied to shoals that fusion and fission across time. STAP incorporates into AP a soft temporal constraint that takes cluster dynamics into account, encouraging partitions obtained at successive time steps to be consistent with each other. We explore how STAP performs under different settings of its parameters (strength of the temporal constraint, preferences, and distance metric) by applying the algorithm to simulated sequences of collective coordinated motion. We study the validity of STAP by comparing its results to partitioning of the same data obtained from human observers in a controlled experiment. We observe that, under specific circumstances, AP yields partitions that agree quite closely with the ones made by human observers. We conclude that using the STAP algorithm with appropriate parameter settings is an appealing approach for detecting shoal fusion–fission dynamics.

1. Introduction

Determining group membership is a main concern of scientists studying animal collective behavior. Despite the growing body of literature devoted to the subject, criteria for determining shoal membership are still a matter of discussion. In particular, there is a need for methods that determine which fish are in a shoal and which are not. A common criterion to determine members of a shoal is based on body length based distances: animals within a criterion distance are considered to belong to the same shoal or group (in fish, four body lengths [1,2]; in dolphins, 100 m [3]). However, given the poor empirical evidence for validating that criterion, determining the limits of the shoal by eye is also common [4].

In this article, we follow the definition of a shoal given by [5], which permits quantification of this behavior, and is based on an earlier definition by [2]. According to [5, p. 614], a shoal is a group of individuals "presenting significant degree of cohesion, limited in a relatively small portion of space, a consequence of social interaction between these individuals". However, defining a shoal, which is a particular kind of social interaction, as being a consequence of social interaction itself, is a circular definition; thus, "definitions based on the geometrical or statistical distributions of individuals (...) are useful in the study of aggregation behaviours because they are operationally objective and are independent of such behaviours" [6, p. 487]. Therefore, an appropriate way to define shoals is to provide an objective method to quantify cohesion, while characterizing them as consequences of social interaction seems unnecessary.

Our current work focuses on cohesion, or aggregation, and seeks to determine how to objectively characterize shoals. When synchronisation of displacements is measured, it is typically assumed that there exists a single group or shoal; however, when

* Corresponding author. Tel.: +34 933 125 088; fax: +34 934 021 359.
E-mail addresses: vquera@ub.edu (V. Quera), francesc.salvador@ub.edu (F.S. Beltran), inmar@psi.utoronto.ca (I.E. Givoni), ruth.dolado@ub.edu (R. Dolado).

0166-4328/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.bbr.2012.11.031

several shoals are detected they can have different degrees of synchronisation and intra shoal aggregation. In such cases, it is more sensible to measure synchronisation and aggregation for each shoal separately. Therefore, a necessary first step in the analysis of collective motion is that of determining how many subgroups exist, and which individuals belong to each group.

When studying aggregation in groups of fish, three statistical methods have been proposed for determining which individuals belong to a shoal and which are outsiders. Such methods aim to detect a main cluster or shoal, and exclude outliers. These methods are based on momentary mean distances among individuals (Miller and Gerlai's 2008 method, MG08 for short, [7]), on trajectories of nearest neighbor distances during a session (Miller and Gerlai's 2011 method, MG11 for short, [8]), and on momentary nearest neighbor distances (Quera, Beltran and Dolado's 2011 method, QBD11 for short, [9]). When the MG08 is applied over several successive time units, the assignment of individuals to the main cluster tends to be unstable in certain cases: as individuals move, some of them may be considered members of the shoal at a certain time unit and excluded from it at the next, while for a human observer no substantial differences in membership can be distinguished. MG11 and QBD11, on the other hand, provide more stable results across time units, though MG11 can only be applied once all the data from a session have been gathered. None of these methods aims to segment a group into more than two disjoint subgroups or shoals. MG11 does segment a group into more than two subgroups, but the subgroups are not necessarily disjoint. A comparative summary of the three methods is shown in Table 1.

Animal group membership can be highly variable, as groups fuse and fission [10]. Regarding the analysis of animal aggregations, it is accepted that determining objectively the number of groups present and which individuals are members of each group is a very difficult task [6]; consequently, a quantitative method is needed to detect multiple groups. Moreover, in order to determine the validity of such a method, its outcome should be compared to estimates made by human observers, which would "ensure consistency and objectivity across time and among different observers and studies" [6, p. 483].

In this article we propose using the affinity propagation (AP) clustering algorithm [11–14] for detecting multiple disjoint shoals. We explore how AP performs under a variety of circumstances, and we compare its output to partitioning results obtained from human observers asked to perform the same task in a controlled experiment. In addition, we propose a variant of the algorithm (denoted by STAP) that takes into account cluster dynamics so that the results are temporally consistent: that is, STAP can yield partitions at time t that are related to those obtained at time $t-1$. This is in contrast to performing standard AP on each time step, yielding independent and not necessarily consistent, results. Unlike MG11, AP can be applied momentarily and not post hoc.

2. Affinity propagation

Cluster analysis, or clustering, is the task of partitioning data into disjoint subsets or groups. When groups also need to be associated with a label, the task is known as classification. There exist a wide variety of clustering and classification techniques, such as hierarchical cluster analysis, nearest neighbor classification, and techniques based on swarm intelligence algorithms (e.g., [15,16]). An iterative, kth nearest neighbor, hierarchical cluster analysis for detecting shoals was been proposed in [6]. However, results from hierarchical cluster analysis require a cutoff criteria in order to determine the actual clusters. k-Means clustering has been proposed for detecting clusters in animal social networks [17], but, similarly to many other methods, it requires specifying the number of clusters a priori. In comparison, affinity propagation partitions data into clusters without requiring a cutoff criteria or knowing the number of clusters to find. However, like other data clustering methods, AP requires that some parameters must be specified by the users; in this article we explore systematically how those parameters affect AP performance. Affinity propagation was developed by Delbert Dueck and Brendan J. Frey and has become a popular method in many research fields such as machine learning, bioinformatics, social networks analysis, computer vision, and neuroscience [14, pp. 6–7]. In this article we present an overview of AP; for details, we refer the reader to [11–14,18].

The affinity propagation algorithm takes as input a matrix of pairwise similarities for n points (s_{ij}, $i=1\ldots n$, $j=1\ldots n$). In our case, similarities are defined between individual fish represented by their spatial coordinates. The algorithm partitions the points into clusters so that each cluster contains exactly one prototypical data point, known as the *exemplar*, to which the other points in the cluster are associated. The similarity s_{ij} is a measure of how suitable point j is to serve as the exemplar for point i; the similarity between a point and itself, s_{jj}, is known as the *preference*, and is a measure of the a priori suitability of point j is to serve as an exemplar.

The algorithm operates by an iterative message-passing mechanism; each data point can be thought of as a node in a network. Nodes send and receive messages to and from other nodes along the edges of the network. Each node i transmits its *responsibility* (ρ_{ij}) for recognizing other nodes j as candidate exemplars, and its *availability* (α_{ij}) to be a candidate exemplar for other nodes and for itself (*self-availability*, α_{jj}). Respectively, these messages reflect the accumulated evidence "for how well-suited point j is to serve as the exemplar for point i, taking into account other potential exemplars for point i" and "for how appropriate it would be for point i to choose point j as its exemplar, taking into account the support from other points that point j should be an exemplar" [13, p. 972]. Message passing is an iterative process in which responsibilities and availabilities are updated as functions of similarities and previous responsibilities and availabilities. After a certain number of iterations the process typically converges, and the messages no longer change between iterations. At that point, the messages can be used to compute the subset of points that are the exemplars, as well as

Table 1
Comparative summary of existing methods for detecting one single shoal vs. outliers. MG08, [7]; MG11, [8]; QBD11, [9].

	Methods		
	MG08	MG11	QBD11
Measure	Mean inter-individual distances	kth nearest neighbor distances	kth nearest neighbor distances
Analysis	Momentary	Post hoc	Momentary
Segmentation criterion	Arbitrary (square root of global mean distance)	Distribution-based and arbitrary (percentile)	Arbitrary (square root of global mean distance, and percentile)
Result	Main cluster/outliers	Main clusters/"k-type excursions"	Main cluster/outliers
Size of main cluster	Liberal	More conservative as total group size increases	Conservative
Stability of segmentations	Fairly unstable	Fairly stable	Moderately stable
Computational cost	Low	High	Low

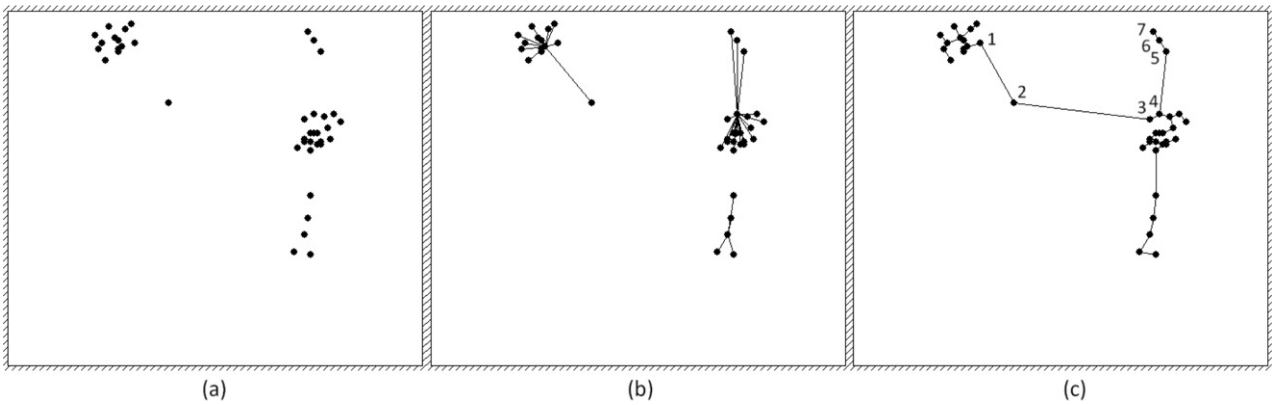

Fig. 1. An example of the affinity propagation algorithm. (a) Data points from a snapshot of our program MovAgent, which simulates coordinated collective motion [20]. (b) Clusters found by the algorithm, with each exemplar point connected to the other points in its cluster. (c) Minimum spanning tree connecting the data points; the geodesic distance between points 1 and 7 is the total length of the segments connecting 1-2, 2-3, 3-4, 4-5, 5-6, and 6-7, and is greater than their Euclidean distance; geodesic distances tend to maximize between-cluster distances.

the association of every other data points to an exemplar. It can be shown that AP is an approximate method for finding exemplars such that the sum of similarities between the data points and their exemplars is maximized.

While AP *finds* exemplars in the data points themselves, methods like *k*-means clustering *generate* exemplars, which are averages of the data points. Thus it is unlikely that a data point becomes an exemplar itself. Also, contrary to *k*-means and other methods, AP does not require the number of clusters to be specified a priori; in AP, the number of clusters that finally emerge depends on the preference: the greater the preference, the greater the number of clusters that tend to emerge [11,13]. Given a matrix of similarities, the range of sensible preference values is bounded by a maximum and a minimum preference, which are the values that cluster the n points into n clusters or into a single cluster, respectively; these bounds can be computed as functions of maximal values in the similarity matrix [19].

When points represent spatial positions, as in the case considered here, similarities can be computed from their distances d_{ij}. For example, $s_{ij} = -d_{ij}^2$ [13], and thus all similarities take negative values; the closer to zero the higher the similarity. Usually, the preference is set to some percentile of the distribution of similarities among the points. The AP algorithm starts by setting all availabilities to zero, $\alpha_{ij} \leftarrow 0$. In each iteration, responsibilities and availabilities are updated as follows:

$$\rho_{ij} \leftarrow s_{ij} - \max_{k \neq j}\{\alpha_{ik} + s_{ik}\} \tag{1}$$

$$\alpha_{ij} \leftarrow \min\{0, \rho_{jj} + \sum_{k \neq i,j} \max\{0, \rho_{kj}\}\} \tag{2}$$

$$\alpha_{jj} \leftarrow \sum_{k \neq i} \max\{0, \rho_{ki}\} \tag{3}$$

The quantities are computed iteratively until α_{ij} and ρ_{ij} do not vary between iterations. Upon convergence, each point k that satisfies $\alpha_{kk} + \rho_{kk} > 0$ is deemed to be an exemplar. Fig. 1 shows a set of 40 data points and how AP partitions them into three disjoint clusters.

Usually, similarities are converted from Euclidean distances. However, AP can fail when the data points form structures that are linearly non-separable. In that case using geodesic distances appears to be preferable [21]. The geodesic distance is a graph-based metric. The distance between two data points is the length of the shortest path connecting the points when traveling along graph edges. In image processing, the geodesic metric has been proposed as an appropriate alternative to the Euclidean one for describing the topological structure of groups of pixels because measuring distance based on shortest paths better represents connectivity [22]. Given data points and Euclidean distance, a graph can be constructed in multiple ways, and we have chosen to use the *minimum spanning tree*. Given the point coordinates, we first compute the minimum spanning tree (for example, using Prim's algorithm [23]), and then compute for every pair of points the geodesic distance by summing the lengths of the edges along the only path connecting the points (using Floyd's algorithm [24]). Fig. 1c shows the minimum spanning tree for the data points in Fig. 1a.

3. Affinity propagation with a soft temporal constraint (STAP)

When AP is applied independently at successive time steps to points in motion, some pairs of points that, according to AP, belong in the same cluster at time step t may be assigned to two different clusters at $t+1$. They can then be assigned yet again to the same cluster at $t+2$. Visual inspection of the data may sometimes suggest that the points should remain together in the same cluster throughout these time-steps. Several extensions to AP have been proposed in which constraints are included in order to enforce some dependency of clustering solutions over time [25–27].

Here, we derive a variation of the AP algorithm that includes a soft temporal constraint when applied at successive time steps. We denote this variant by STAP. We use the idea of hard constraints along time introduced in [27], in which points that were assigned to the same cluster at some time step were assigned mandatorily to the same cluster at subsequent time steps. However, we relax this constraint to be a soft constraint so that clusters can break; using a hard temporal constraint would force the algorithm to maintain the same clusters at successive time steps, thus preventing the detection of possible shoal fissions and fusions. We used the binary variable model for AP, proposed by [18], which permits deriving variations of the algorithm using a factor graph [28]. We omit the details of the derivation and present here final formulas for updating the availabilities when the temporal constraint is taken into account.

Let us denote by G_j^{t-1} the group of points that were in the same cluster as point j in the previous time step; if i and j were in the same cluster at $t-1$, then $G_i^{t-1} = G_j^{t-1}$. Also, let us denote by c a non-negative soft temporal constraint that reinforces assigning two points to the same exemplar if in the previous time step both points were in the same cluster (whose exemplar was either point, or a third one). Note that the setting $c = 0$ degenerates to regular

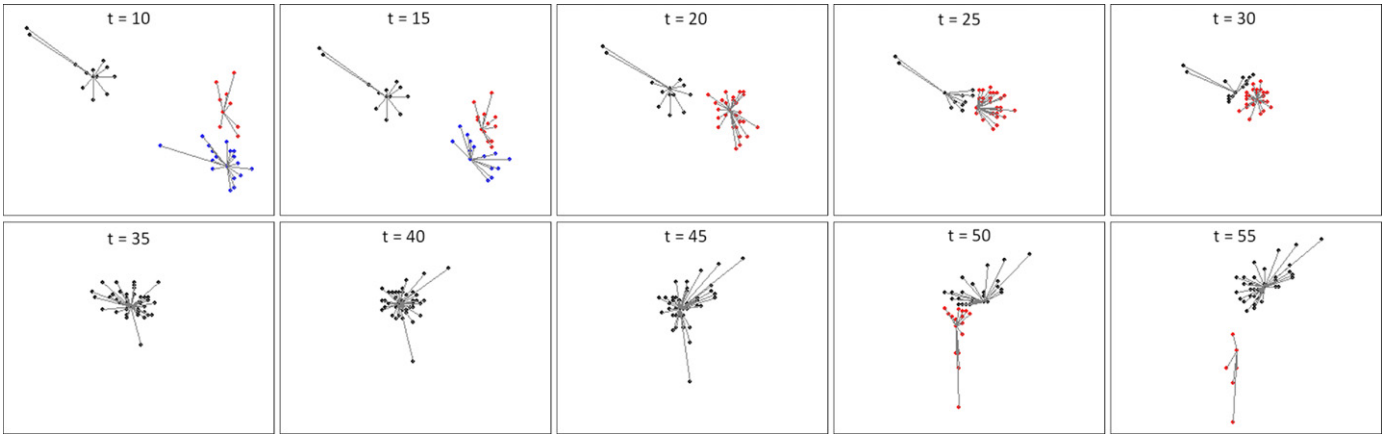

Fig. 2. An example of how STAP (AP with a soft temporal constraint) detects subgroup fusion and fission. Results shown were sampled from a sequence every five time steps. Three clusters were found at $t = 10$ and $t = 15$, two clusters at $t = 20$ through 30; one cluster at $t = 35$ through 45; and two clusters at $t = 50$ and 55. Exemplar points are connected to the other points in their clusters. The temporal constraint is set to $c = 4000$, the preference is set equal to the median of the distribution of similarities, and the geodesic distance metric was used to compute similarities. Each cluster is represented by a different color.

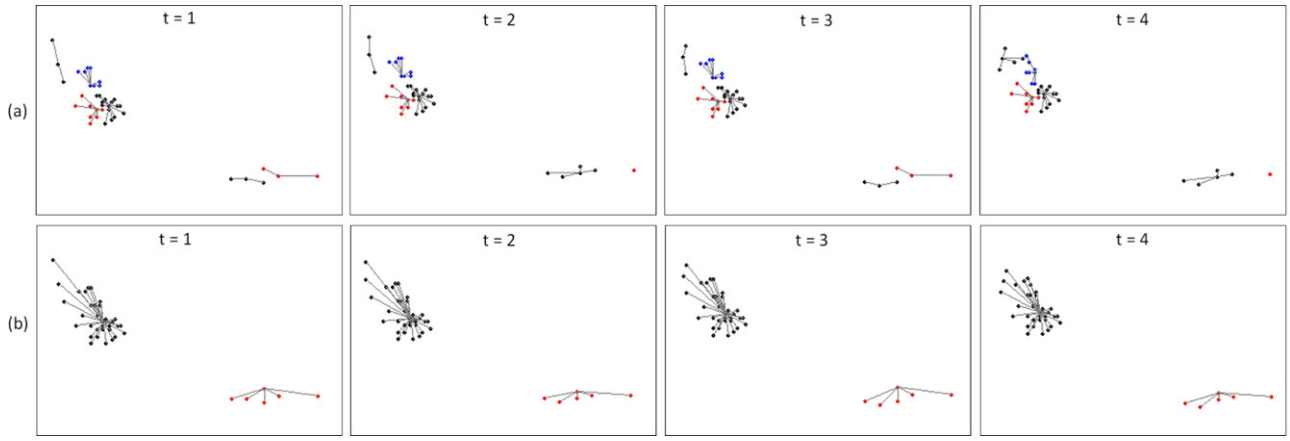

Fig. 3. A comparison between affinity propagation without and with a soft temporal constraint applied to a group of 40 points in motion for four consecutive time steps, from a MovAgent simulation. (a) Regular AP is used, yielding six clusters at each time step; note that some points in the lower right that were in different clusters at $t = 1$ are assigned to the same cluster at $t = 2$, to different clusters at $t = 3$, and again to the same cluster at $t = 4$. (b) AP with a soft temporal constraint is used ($c = 4000$), yielding two temporally consistent clusters at the four time steps; the preference was set equal to the median of the distribution of similarities, and the geodesic distance metric was used to compute similarities. In both cases, exemplar points are connected to the other points in their clusters. Each cluster is represented by a different color.

AP [13]. Based on the binary variable model, we derive the following availability updates, which now depend on the temporal constraint:

For the case $i = j$:

$$\alpha_{jj} \leftarrow \max\{-c + \sigma_{ij}, \tau_{ij}\} \tag{4}$$

For the case $i \neq j$ and $i \notin G_j^{t-1}$:

$$\alpha_{ij} \leftarrow \min\{0, \max(-c + \rho_{jj} + \sigma_{ij}, \rho_{jj} + \tau_{ij})\} \tag{5}$$

For the case $i \neq j$ and $i \in G_j^{t-1}$:

$$\alpha_{ij} \leftarrow \max\{-c + \rho_{jj} + \sigma_{ij}, \rho_{jj} + \tau_{ij}\} - \max\{0, -c + \rho_{jj} + \sigma_{ij}\} \tag{6}$$

where:

$$\sigma_{ij} = \sum_{k \neq i,j} \max\{0, \rho_{kj}\} \tag{7}$$

$$\tau_{ij} = \sum_{\substack{k \neq i,j \\ k \in G_j^{t-1}}} \rho_{kj} + \sum_{\substack{k \neq i,j \\ k \notin G_j^{t-1}}} \max\{0, \rho_{kj}\} \tag{8}$$

Note that as it turns out, responsibilities in this model are updated as in regular AP. As can be seen, the temporal constraint c is subtracted from the availabilities and responsibilities, which in turn are functions of the similarities among data points. Therefore, in order to have effect on the clustering process, the temporal constraint should be assigned a value on the same scale as the square of the distances among the data points. Fig. 2 shows an example of subgroup fusion and fission as detected by AP with the soft temporal constraint.

Fig. 3 demonstrates how STAP is affected by the value assigned to temporal constraint c. It being a constraint, we expect that the greater its value the more similar the partitions obtained when the algorithm is applied at successive time steps to a group of points in motion whose speed is moderate[1]. We are also interested in the overall relationships between the temporal constraint values, and the two additional parameters of the algorithm: the preference values, and the choice of distance metric. As mentioned before, it is expected that the greater the preference the greater the number of clusters detected by AP at a given time step; we are interested in exploring how varying the preference actually affects the number

[1] In contrast, if points move at high speed and their positions are sampled at large intervals, the partitions obtained at successive intervals may not be necessarily similar.

of clusters, and whether this effect is independent or not from the temporal constraint in STAP. Regarding the similarities, we are interested in investigating which distance metric, Euclidean or geodesic, can yield partitions that are more similar to how human observers would partition the data points.

Our objective is twofold: first, we wish to test the performance of STAP on several sequences of data points by exploring different combinations of the parameter settings (varying temporal constraint strength, preference strength, and choice of distance metric). Second, we wish to evaluate how STAP performs by comparing its results using a variety of parameter settings with those obtained by human observers who are asked to partition the same groups of data points in a controlled task. Only through such experimental comparison we can conclude how well the STAP clustering method partitions data.

4. Experiments with affinity propagation

Using our agent-based simulation software MovAgent (http://www.ub.edu/gcai/movagent), we simulated several sequences of coordinated motion in a group of 40 virtual fish, which moved on a bounded two dimensional lattice of size 120×120. Each fish is represented by a point occupying a lattice cell at each time step. The program makes coordinated motion emerge in a group from initial random movements by applying local interaction rules among the agents that make them adapt their movements to each other's; coordination can be achieved at a global level. Alternatively, several subgroups, each with its own coordination, can emerge, split, or fusion. Usually, coordinated motion can be seen on the computer screen after several hundreds of time steps. The rules are similar but not identical to those in Huth & Wissel's shoaling and schooling model [29]; for details about MovAgent, see [20,9]. Like Huth & Wissel's model, the graphical display of the simulated motion is perceived to be similar to that of real fish.

We selected ten fragments (named A, B, . . .J, ranging from 83 to 159 time steps, with 140.5 consecutive time steps on average) from the simulated sequences and extracted the two-dimensional coordinates of each virtual fish at every time step; the onset of each fragment corresponded to a time step when coordinated motion had already emerged. The criterion to select the fragments was that the sequence dynamics exhibited both group fusion and fission. We developed APAsoft, a computer program written in C language for Windows, that reads the two dimensional coordinates at successive time steps and applies STAP at every step. The program's performance (with the temporal constraint set to zero) was validated by comparing its results for a sample of time steps with those provided by Frey and Dueck's apcluster C code [13] and by the R apcluster package [30], both of which implement regular AP, and no differences were found.

For a given segment, APAsoft was run 132 times, once for every combination of the following parameters: (a) temporal constraint $c = 0$, 2000, 4000, 6000, 8000 and 10,000 (6 conditions); (b) preference = minimum similarity, or percentiles 5, 10, 15, . . ., 50 (by 5) of the distribution of similarities (11 conditions); (c) similarities converted from either distance metric, Euclidean or geodesic (2 conditions). At every time step, the similarity between two virtual fish was computed as minus the (Euclidean or geodesic) squared distance between them; given the size of the lattice, the minimum possible similarity was $-2 \cdot 120^2 = -28,800$ when Euclidean distances were used, and lower than it when geodesic distances were; therefore, for the reasons given previously, we assigned the constant values in the thousands.

We observed that in some occasions AP failed to converge when the number of iterations of the algorithm reached the default used in Frey and Dueck's apcluster program (500 iterations). In those cases APAsoft increased the number of iterations and simultaneously the *damping factor*, a variable that moderates the updating of availabilities and responsibilities at each iteration of the algorithm in order to prevent sudden oscillations [13]. We also observed that the difference in the number of clusters found by AP in two successive time steps was in some cases disproportionate, while visual inspection suggested that the change in number of clusters should be much smoother; to avoid that, an upper admissible limit L_t was defined for the number of clusters at t, N_t, given the number of clusters found at $t-1$, N_{t-1}: $L_t = 2 + N_{t-1} + N_{t-1} \bmod 2$; if $N_t > L_t$, then the partition found by AP at t was ignored, and the data points were assigned to the same clusters as in $t-1$. For two sample fragments AP was applied 41,448 times (314 time units in total $\times 132$ runs) under the different conditions, $N_t > L_t$ occurring only in 105 instances (0.253%).

4.1. Comparing clusterings

Results obtained when STAP was applied to the same fragment while varying the parameters were compared using Hamming distances. The Hamming distance [31] is a measure of how different are two code sequences of the same length for which a point to point correspondence can be assumed. It is defined as the number of positions where codes in one sequence differ from codes in the other; for example, the Hamming distance between sequences `01001001111001000101` and `11000101101001010100` (length 20) is $HD = 6$, and their normalized Hamming distance is $NHD = 6/20 = 0.3$. Many distance and similarity indices have been proposed for comparing results obtained using different clustering methods (e.g., [32–36]). The statistical properties of many of those indices have been studied, and equivalences between some of them have been found [37,38]. In particular, the NHD between two clusterings equals one minus the Rand index [32], a similarity measure commonly used in cluster validation [37,39]; a correction for chance has been proposed for the Rand index [34], and therefore for the NHD, but our interest is using the latter as a simple descriptor without reference to its probability distribution.

In order to compute the Hamming distance between alternative clustering solutions of the same n data points obtained by STAP under two different conditions, we must know how many pairs of points were assigned to the same cluster in both cases (n_{11}), how many pairs were assigned to the same cluster in one case but to different clusters in the other ($n_{10} + n_{01}$) and how many pairs were assigned to different clusters in both cases (n_{00}). The Hamming distance between the two clustering solutions is then $HD = n_{10} + n_{01}$, and the normalized distance is obtained by dividing it by the total number of pairs of points, $\binom{n}{2} = n_{11} + n_{10} + n_{01} + n_{00}$, that is, $NHD = (n_{10} + n_{01}) / \binom{n}{2}$. Obviously, if the two clusterings are identical, their distance is equal to zero. The NHD can attain its maximum (1) only when in the first clustering all points are together in a single cluster and in the second clustering every point is a cluster, or vice versa; in that case, $n_{10} = \binom{n}{2}$.

For example, suppose that, given $n = 6$ data points, two different clustering solutions were obtained, with two and three clusters, respectively. Cells in Table 2 contain labels indicating the cluster number to which points were assigned. In Table 3, pairs of points sharing cluster labels are represented by ones in the triangular matrices and pairs of points belonging to different clusters are represented by zeroes. The Hamming distance between the two clusterings is obtained by counting how many cells in the left matrix differ from their corresponding cells in the right one; in this

Table 2
A hypothetical example of two different clustering solutions of a group of six data points. Table cells are cluster labels to which the points were assigned.

	Data points					
	1	2	3	4	5	6
Clustering 1	2	2	1	1	1	1
Clustering 2	1	2	2	1	1	3

example, $HD = n_{10} + n_{01} = 9$ and $NHD = 9/15 = 0.6$ (number of pairs is $\binom{6}{2} = 15$).

4.2. STAP results

Fig. 4a shows how varying the temporal constraint and the preference affect the number of clusters that are detected by STAP, using either Euclidean or geodesic distances among the virtual fish for computing their similarities. When no temporal constraint is applied ($c = 0$, regular AP) the higher the preference, the greater the number of detected clusters using either distance metric. This result is in accordance with what was predicted by [13]. In general, using the Euclidean metric tends to yield more clusters than using the geodesic one irrespective of the preference; only for fragment I the number of clusters detected does not seem to depend on the metric. When a temporal constraint is applied, the effect of the preference values on the number of clusters detected is greatly attenuated, a result that holds for the ten fragments we analyzed.

Table 3
Binary matrices indicating which pairs of points belonged to the same cluster (1) or to different clusters (0) according to the two clustering solutions defined in Table 2.

	Clustering 1				
2	3	4	5	6	
1	1	0	0	0	0
2		0	0	0	0
3			1	1	1
4				1	1
5					1

	Clustering 2				
	2	3	4	5	6
1	0	0	1	1	0
2		1	0	0	0
3			0	0	0
4				1	0
5					0

The number of clusters tends to increase only when the preference value approaches the median, and not by as much as in the absence of the temporal constraint.

Although replications with more fragments would be necessary in order to generalize these results, they indicate that the temporal constraint tends to neutralize the effect of the preference on the number of clusters. In other words, deciding which preference to use is largely irrelevant when the temporal constraint is applied. Fig. 4b shows that the number of clusters detected by STAP at

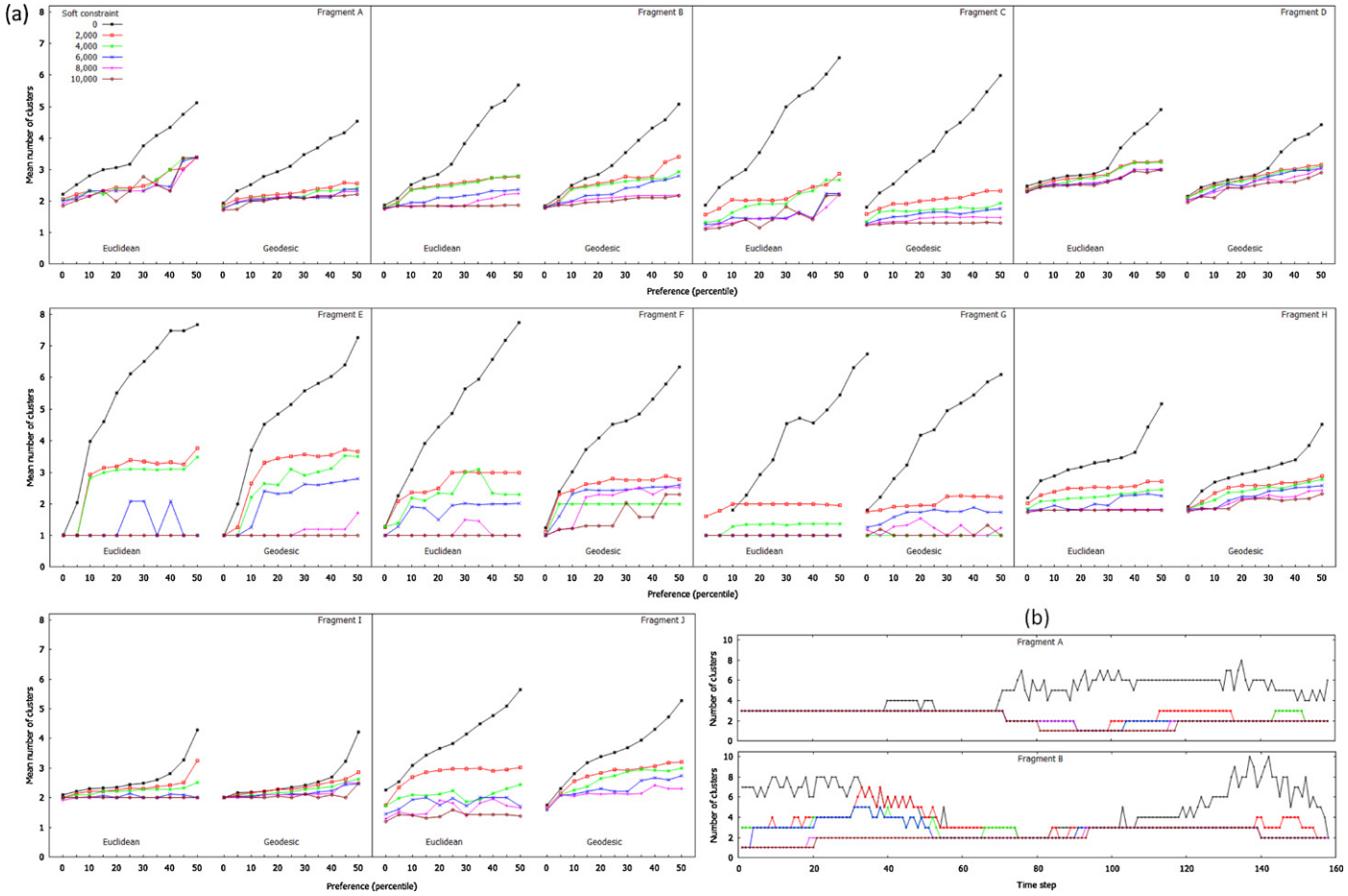

Fig. 4. (a) Mean number of clusters detected by STAP in ten fragments (A through J), for all parameter settings of the soft temporal constraint c, the preference, and the distance metric. Preference 0 stands for the minimum similarity among data points. (b) Number of clusters detected in fragments A and B when different values of the soft temporal constraint are used (preference = 50, geodesic metric).

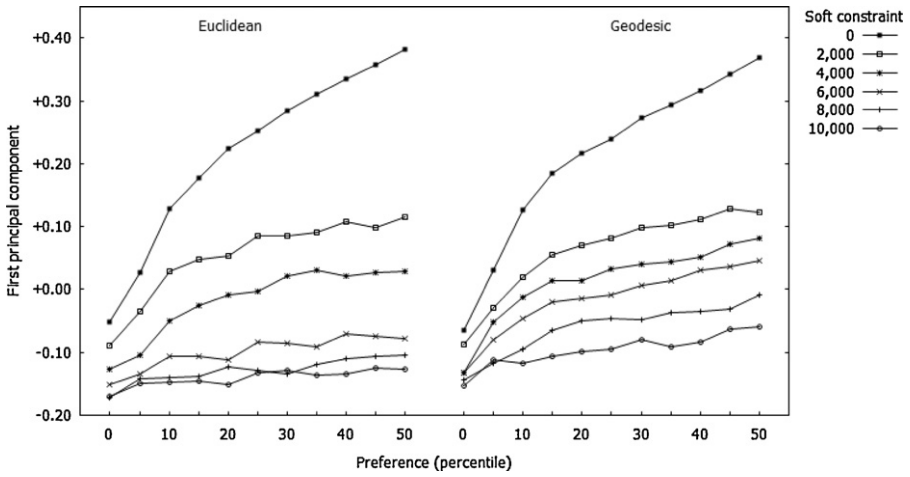

Fig. 5. Coordinates of the first component from a principal component analysis on a global, averaged symmetrical matrix of mean Hamming distances between the STAP clustering results of all different parameter settings; distances measured along that component represent distances between clustering solutions. Results are shown for all parameter settings of the soft temporal constraint c, the preference, and the distance metric; preference 0 stands for the minimum similarity among data points. Solutions that are equivalent have close values on the vertical axis (e.g., $c = 4000$, Euclidean metric; preferences = 45 and 50). See text for explanation.

successive time units in fragments A and B is considerably less stable when $c = 0$ than otherwise; similar results were found for other combinations of preference values and distance metrics.

For every time step within a fragment, we computed NHDs among the STAP solutions for each of the 132 parameter value settings. The NHDs were averaged for all the time steps of a fragment, and arranged in a 132×132 symmetrical matrix, one per fragment; a global 132×132 matrix was then obtained by averaging the ten matrices. A principal component analysis of the global matrix was performed using R's cmdscale and setting the maximum number of dimensions of the space to two. The first two eigenvalues obtained were 2.355 and 0.240, indicating that distances along the first component adequately represent the observed NHDs between clusterings in both fragments. Therefore, in Fig. 5 we present the coordinates of only the first component for the 132 combinations of soft temporal constraint, preference value, and distance metric.

Points with similar or identical coordinates indicate that the partitions yielded by STAP given their associated parameter settings are very similar or identical on average. For example, when STAP was applied with the temporal constraint $c = 4000$ using Euclidean distance, the same or very similar partitions were obtained with a preference setting of 40, 45 or 50, on average. Therefore, the points that represent those parameter settings have identical or nearly identical coordinates in the first component. Clusterings found when the preference is small are very similar across different values of the temporal constraint; however, as the preference approaches 50, results found under different values of the constraint are more similar when $c > 0$ but different from those when $c = 0$. Thus, using a non-zero temporal constraint tends to make clusterings dependent only on the preference and the distance metric. When $c = 0$, increasing the preference has a clear effect on the clusterings, for both Euclidean and geodesic distances: very different results are obtained when the preference is equal to the minimum similarity than when it is equal to the median similarity. On average, across all fragments, the effect of the preference is attenuated when the temporal constraint is increased. These results are consistent with the results for the mean number of clusters (Fig. 4a).

If the soft temporal constraint has the desired effect on the clusterings detected by STAP, then the NHD between clusterings obtained at time steps t and $t + 1$ should be smaller when the temporal constraint is big as opposed to when it is small or zero. That is precisely what the results in Fig. 6 indicate. In that figure, mean NHDs between clusterings detected at consecutive time steps, averaged for the ten fragments analyzed, are represented as a function of temporal constraint, preference and distance metric. Mean NHDs between results at two consecutive time steps decrease

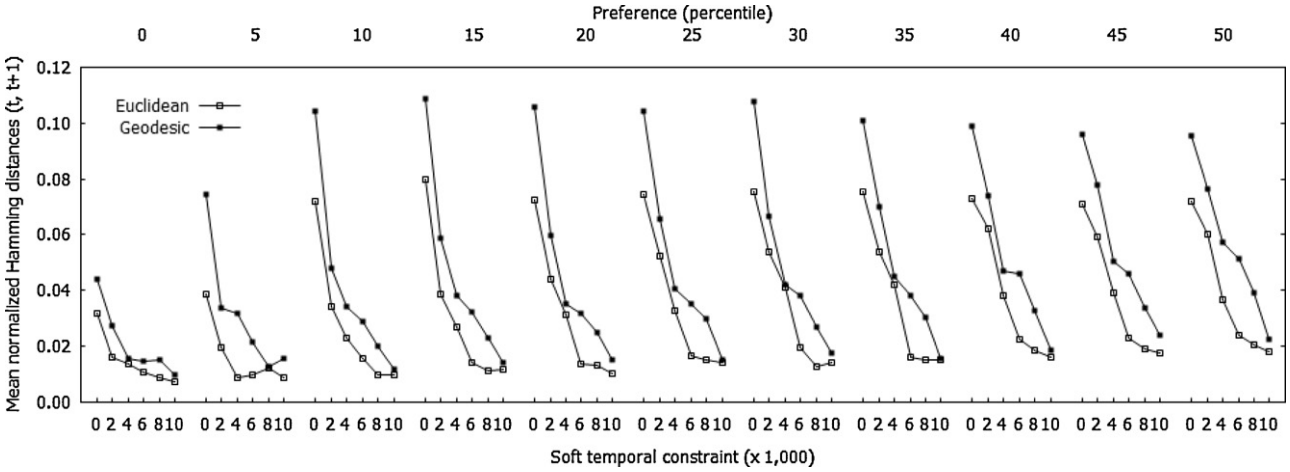

Fig. 6. Mean normalized Hamming distance between STAP clusterings at t and $t + 1$, averaged for ten fragments, for all parameter settings of the soft temporal constraint c, the preference, and the distance metric. Preference 0 stands for the minimum similarity among data points.

drastically as the temporal constraint is increased, compared to $c = 0$. This decrease is more clear and systematic when geodesic distances are used. Therefore, when the temporal constraint is applied, the clusterings obtained by STAP in two successive time units tend to be similar, whereas when $c = 0$ there are often considerable differences between the clusterings. We conclude that when the objective is detecting fusions and fissions of shoals that occur gradually, STAP proves to be a useful algorithm.

5. Comparing affinity propagation clustering with human clustering

Analyzing the quality of clustering results is known as *cluster validation*, which can be internal (how well the results fit the data), external (how well they fit a pre-specified structure) or relative (how well they fit results from other clustering methods) [40]. According to [6], objectivity of the clustering results can be ensured by comparing them with how human observers partition the same data. Thus, by comparing to human observers, we obtain a type of relative validation in which human performance is considered as if it were an alternative clustering method.

We cannot assume, however, that all human observers will agree about partitioning when presented the same data sets, and therefore a sample of observers is necessary. In this section we describe an experiment in which samples of the data points already used in Section 4 were presented to human participants. Partitioning results obtained by STAP and by humans on same data were compared using normalized Hamming distances.

5.1. Participants

Thirty-four participants (eight men and 26 women) volunteered for the experiment ($M = 23.18$, $SD = 3.83$, age range: 19–33 years). Twenty-five participants were undergraduates and nine were graduate students from the Department of Psychology at the University of Barcelona. The participants were recruited at an ordinary class session.

5.2. Material

Fragments A and B, which were previously analyzed using APASoft (see Section 4), were sampled regularly every six time steps. Two subsets of two-dimensional coordinates data were obtained (fragments A' and B'), 27 time steps each. Images representing the coordinates at each time step were displayed on a computer screen as experimental stimuli by the program Segment [41], written in C#.NET 3.5 and running on Windows. The program displayed the images on a 19-in. touch screen according to a predefined experimental plan, and recorded the participants' responses. A snapshot of a stimulus used in the experiment is shown in Fig. 7a.

In order to generate training stimuli we selected an additional fragment 40 time steps long from a sequence of coordinated motion exhibiting group fission and fusion, simulated by MovAgent. We regularly sampled the fragment every six time steps, obtaining a total of eight training images.

5.3. Procedure

Participants were randomly divided into two groups. In one group (18 participants) the images for both fragments A' and B' were displayed sequentially in temporal order (sequential group), while for the other group (16 participants) images were displayed at random (random group). This division allowed us to determine whether presenting the images sequentially or at random could have an effect on how points were partitioned. All participants observed all the images from the two fragments, 54 images in total. Thus, a two-factor mixed experimental design was used.

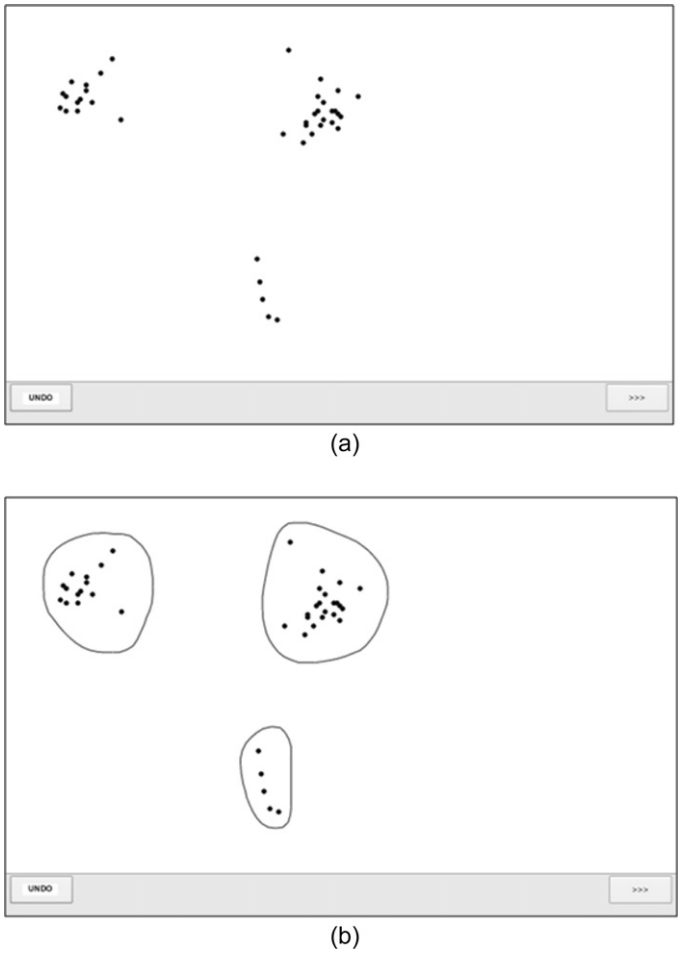

Fig. 7. (a) Snapshot of a stimulus displayed on a touch screen and (b) an example of how a participant encircled the points.

Each participant was tested individually. The experimental procedure was as follows. First, the Segment program displayed text with instructions on the touch screen. The instructions informed the participant that several images containing black points were to follow. The task was to encircle points that made up a group, according to the participant. One or several groups of points could be encircled. The participant simply drew a circle with her or his finger on the screen and the program displayed it as a red line (see Fig. 7b). It was also possible to correct the response by clicking on a *undo* button located at the bottom left of the screen. In such cases, the circles drawn by the participant on the current image were erased, and the participant could try again. Once the task for each image was completed, the participant touched a button on the bottom right on the screen, and the next image was shown.

The instructions emphasized that each point could only be assigned to a single group, that is, groups should be disjoint. In addition, participants were asked to assign every point to some group, where groups were allowed to contain a single point. All participants performed the task with the eight training images. After training, the experimenter asked the participants whether they had any questions, and made sure that they had understood the task correctly. Then, the participants performed the task first with the 27 experimental images for one of the fragments, and then with the 27 experimental images for the other fragment. A blank screen was displayed for three seconds following the first 27 images, indicating that the images for the first fragment had

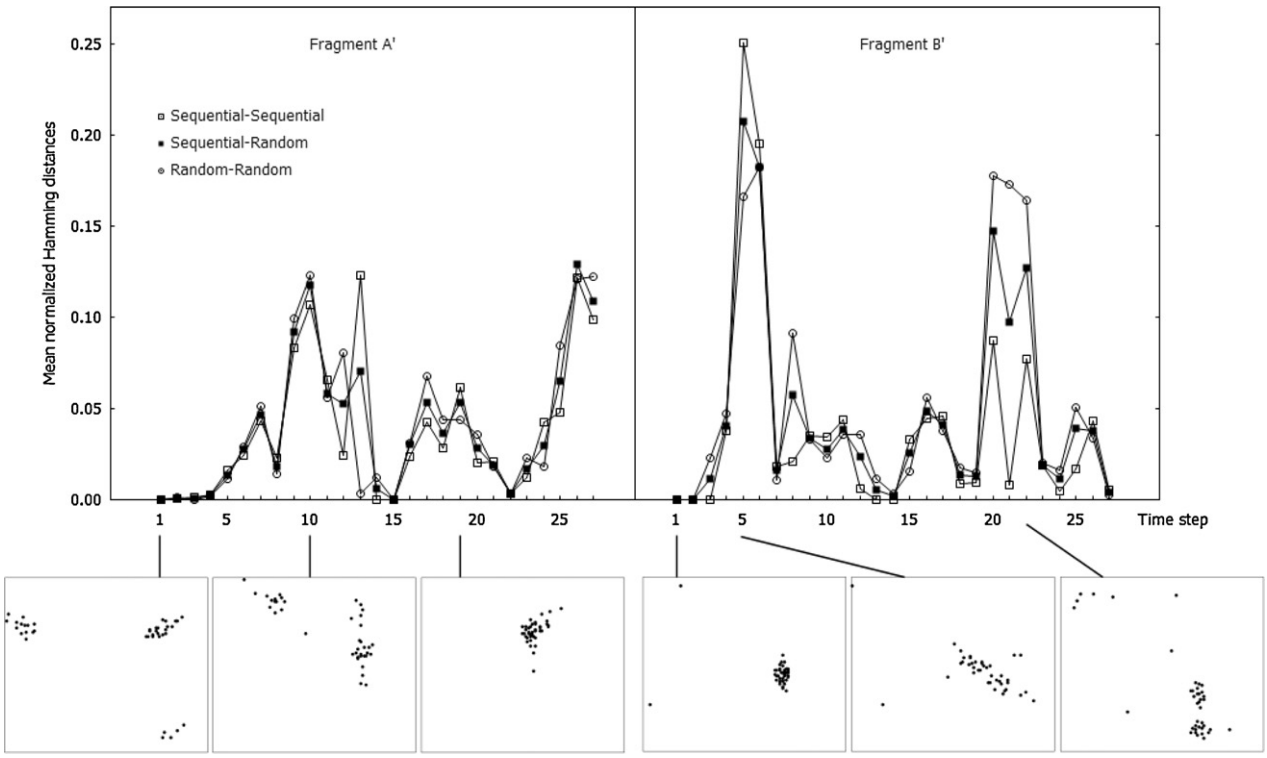

Fig. 8. Mean NHD (normalized Hamming distance) between clusterings for each stimulus in fragments A' and B' according to the participants. Results for between and within group comparisons are shown. Peaks in the graphs indicate disagreement among the participants. Three stimuli about which the participants had various levels of agreement are shown for each fragment.

been completed. The order in which the two fragments were presented was counterbalanced across the participants; that is, in each group of participants half performed the task first with fragment A' then with fragment B', and the other half performed it first with fragment B' then with fragment A'. For each image, the program recorded which groups of points were encircled by the participant, and automatically assigned them the same label. Thus, results for each time step within a fragment consisted of a labels vector with as many elements as the number of points (40). The labels identified the groups to which elements belonged according to the participant.

5.4. Results

In order to compare the participants' performance with that of the STAP algorithm, we must first determine whether there exists a substantial agreement on partitioning results among the human participants. In cases where such agreement was found, we used the participants' responses as a criterion for evaluating STAP. Stimuli for which there was little to no agreement were discarded.

For both fragments, we averaged the NHDs over all pairs of participants for each of the 27 stimuli. Results are shown in Fig. 8 separately for within and between group comparisons; peaks in the figure indicate participants' disagreement on partitioning. As the graph indicates, for fragment A' agreements (and disagreements) were fairly similar regardless of the group to which the participants had been assigned. In other words, displaying the images sequentially or at random was largely inconsequential as far as agreement or disagreement is concerned.

In fragment A', there was a high agreement about how to partition stimuli 1 through 4; then, disagreement increased until stimulus 10, decreased until stimulus 15, and so on. In fragment B', very low agreement was observed among the participants for stimuli 5, 6, 20, 21 and 22, for which the mean NHD was much larger than for any other stimuli (greater than 0.15 in at least one of the between and within group comparisons). For this reason, we discarded the data of these five stimuli from subsequent analyses. Six snapshots of the stimuli in fragments A' and B' can be seen at the bottom left of Fig. 8 for various levels of agreement.

For each of the remaining stimuli, we computed the mean NHDs between the clusterings according to the participants, and those obtained by STAP under different combinations of the parameters (soft temporal constraint = 0, 2000, 4000, 6000, 8000, 10,000; preference = minimum similarity, percentiles 25, 50; metric = Euclidean, geodesic). Given the fact that the mean NHDs for both between and within group comparisons were similar, as Fig. 8 indicates, we pooled the data over the two groups, for each fragment separately. All 27 stimuli in fragment A' and 22 stimuli in fragment B' were used in this analysis. Results are shown in Fig. 9.

For both fragments, when the preference used in STAP equals the minimum similarity among points, varying the temporal constraint has no remarkable effect on the distance between the participants' and the algorithm's clusterings. However, when the preference is set equal to the 25th or 50th percentiles of the similarity distribution, using a non-zero temporal constraint highly decreases the distance when compared with $c = 0$, and more so when similarities are computed using the geodesic metric. When the temporal constraint is zero, the greater the preference the greater the mean distance between the participants' and the algorithm's clusterings. On the other hand, when the preference is at the minimum, the type of metric used has little effect on the mean NHD. However, when a temporal constraint greater than zero is specified and the preference is equal to the 25th or 50th percentiles, the mean NHD is smaller when the geodesic metric is used, an effect that is clearly evident when the preference is equal to the median similarity among points.

In summary, for the fragments we explored, results obtained by STAP are mostly in agreement with how humans

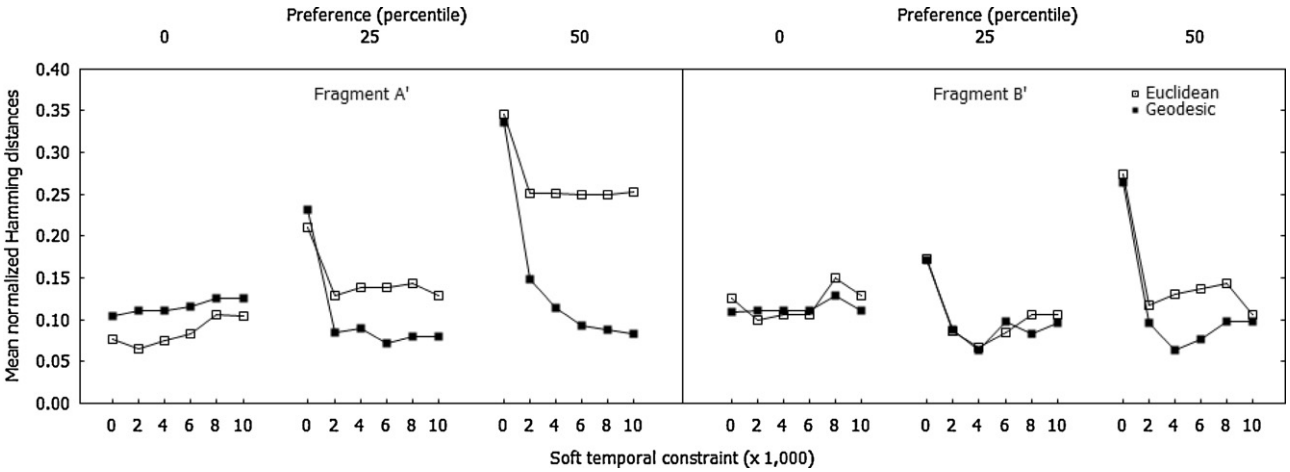

Fig. 9. Mean NHD (normalized Hamming distance) between clusterings according to the participants and clusterings obtained by STAP for combinations of soft temporal constraint (0, 2000, 4000, 6000, 8000 and 10,000), preference (minimum, percentiles 25, 50), and distance metric. Preference 0 stands for the minimum similarity among data points. Results for fragments A' and B' are shown.

(on average) partition the same data when a big temporal constraint, a non-minimum preference and the geodesic distance metric are used. However, when the preference is minimal, some results from fragment A' contradict those from fragment B', because when the Euclidean metric is used the partitions obtained are as close to those obtained by humans as when the preference is non-minimum and the temporal constraint is greater than zero.

6. Discussion

The first objective of our research was to test the performance of affinity propagation with a soft temporal constraint (STAP) by exploring different values for the algorithmic parameters: temporal constraint, preference values, and the choice of distance metric (Euclidean versus geodesic). As expected, results showed that when no temporal constraint was applied, the higher the preference, the greater the number of clusters detected. Moreover, the use of geodesic distances tended to yield a smaller number of clusters compared to Euclidean distances. When a temporal constraint was applied the effect of the preference on the number of clusters was neutralized, and no clear differences were observed between Euclidean and geodesic metrics.

Moreover, a principal component analysis on the average of the NHDs among the results obtained by STAP under the 132 conditions of the parameters was consistent with the results for the mean number of clusters detected. That analysis showed that when no temporal constraint is used, increasing the preference has an effect on the clusterings, regardless of the choice of distance metrics, but that this effect is attenuated as the temporal constraint value increases. Finally, mean NHDs between clusterings at two consecutive time steps drastically decreases when using a temporal constraint compared to no temporal constraint, an effect that is most pronounced when using the geodesic distance metric.

The second objective of the present research was to determine which values of the STAP parameters yield results that best match results obtained by human observers. The mean NHDs over all pairs of participants for each experimental stimuli image showed that a substantial agreement regarding the clusterings exists among human observers in almost all images, regardless of whether images are displayed sequentially or at random (only six images of one of the fragments had to be discarded as a consequence of low agreement). The globally high agreement observed in the human performances allowed us to compare the data obtained in the human observers experiment with the data obtained in the APAsoft simulations. The mean NHDs between the clusterings according to the participants and those of STAP showed that the best agreement between human observers' and STAP's results was obtained when a large value of the temporal constraint and a non-minimum preference were used. Results regarding the best choice for the distance metrics are non-conclusive. Even though in general the agreement observed between human observers and STAP was higher when using geodesic metric, an exception was observed in one of the fragments, where combining minimum preferences and Euclidean metric resulted in an agreement similar to that obtained when combining non-minimum preferences and geodesic metric.

In general, the non-zero values that we chose for the soft temporal constraint (2000, 4000, 6000, 8000 and 10,000) had slight effects on the agreement between results obtained by humans and those obtained by STAP. As mentioned before, values in the thousands were assigned because of the size of the lattice used in the simulations, though it is also possible (and perhaps desirable) to use values with a larger spread. Although further explorations are required, the results suggest that, once an appropriate range of values is decided upon, according to the lattice order of magnitude (tens, hundreds, thousands, etc.) the exact value of the temporal constraint is largely unimportant.

However, some questions still remain to be investigated in future studies. Firstly, in our experiments, we only used two different fragments each containing a series of images. Consequently, our results allow us to achieve only some preliminary conclusions. In order to achieve more definitive results, partitions obtained from STAP should be compared with the performance of human observers on a larger number of fragments showing a variety of fusion and fission dynamics. Regardless, our results do indicate a trend that should be tested more exhaustively.

In addition, if the main objective is to automatically detect which fish belongs to which shoal, then the experimental setting proposed to the human observers should include information both about the individuals' locations (points) and headings (head-to-tail vectors), which would allow the observers to predict their future direction of displacements. Thus, in future experiments, images of real fish should be used, and STAP should be furnished with similarities converted from both distances and differences in polarity among real fish. One option could be using the *F* index of flocking or schooling developed by [42], which is a function of both distances and heading differences between individuals; it is itself a similarity index, which approaches 1 when the individuals are close and have similar headings, and approaches 0 when they are either far away, have

opposite headings, or both. In addition, the data used to test both STAP and human observers should be obtained from dynamics of real shoals. Although the data obtained using our MovAgent simulation program is useful for testing shoal detection methods (see [9,20]), empirical validity based on real shoal data should also be of concern.

Finally, as mentioned previously, in addition to fusion and fission detection, a concern in shoal behavior studies is to determine which individuals belong to a shoal and which are outsiders. A method for detecting fusions and fissions in shoal dynamics should also incorporate a criterion of shoal membership to determine which fish do not belong to any shoal at a given time unit. A starting point could be applying one of the previously proposed statistical methods for detecting outsiders, such as MG08 [7], MG11 [8], or QBD11 [9]. That is, at each time step, outsiders would be first detected by one of those methods, and then STAP would be applied to the rest of the fish that are not considered outsiders. Alternatively, and possibly more preferably, STAP could incorporate the ability to easily create clusters that contain a single member, or detect outsiders. In that case, STAP would be able to detect both shoal fusion and fission and at the same time discard individuals that do not belong to any shoal at a given time step.

6.1. Conclusion

Our results show that applying a soft temporal constraint affects how the STAP algorithm detects shoal membership. Compared to regular AP, which has no temporal constraint, using such a constraint: (a) neutralizes the effect of the preferences, both on the number of shoals that the algorithm detects and on the differences between the clusterings, when non-minimum percentiles are chosen as preferences, and (b) decreases the differences between the clusterings at two consecutive time steps, thus preventing inconsistencies in shoal membership detection across time. In general these effects are clearer when geodesic distances among fish are used.

Regarding clustering validation, the results obtained by STAP agree well with those made by human observers, giving consistency to the results across studies and observers. Thus, when appropriate parameter values are chosen, the use of the STAP algorithm is an appropriate option for detecting fusions and fissions of shoals.

Acknowledgements

This project was supported by grants from the Directorate General for Research of Catalonia (2009SGR-1492) and from the Ministry of Science and Innovation of Spain (PSI2009-09075). The authors thank Salvador Herrando and Olatz López for their help in running experiments and data analysis.

Appendix A. Supplementary Data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.bbr.2012.11.031

References

[1] Pitcher T, Magurran AE, Allan JR. Shifts of behaviour with shoal size in cyprinids. In: Proceedings of the 3rd British freshwater fisheries conference. 1983. p. 220–8.
[2] Pitcher TJ, Parrish JK. Functions of shoaling behavior in teleosts. In: Pitcher TJ, editor. Behaviour of teleost fishes. 2nd ed. Chapman & Hall; 1993. p. 363–440.
[3] Lewis JS, Wartzok D, Heithaus MR. Highly dynamic fission–fusion species can exhibit leadership when traveling. Behavioral Ecology and Sociobiology 2011;65(5):1061–9.
[4] Krause J. Positioning behaviour in fish shoals: a cost–benefit analysis. Journal of Fish Biology 1993;43:309–14.
[5] Delcourt J, Poncin P. Shoals and schools: back to the heuristic definitions and quantitative references. Reviews in Fish Biology and Fisheries 2012;22(3):595–619.
[6] Strauss RE. Cluster analysis and the identification of aggregations. Animal Behaviour 2001;61:481–8.
[7] Miller NY, Gerlai R. Oscillations in shoal cohesion in zebrafish (*Danio rerio*). Behavioural Brain Research 2008;193(1):148–51.
[8] Miller NY, Gerlai R. Redefining membership in animal groups. Behavior Research Methods 2011;43(4):964–70.
[9] Quera V, Beltran FS, Dolado R. Determining shoal membership: a comparison between momentary and trajectory-based methods. Behavioural Brain Research 2011;225(1):363–6.
[10] Sueur C, King AJ, Conradt L, Kerth G, Lusseau D, Mettke-Hofmann C, et al. Collective decision-making and fission–fusion dynamics: a conceptual framework. Oikos 2011;120(11):1608–17.
[11] Dueck D. Affinity propagation: clustering data by passing messages [dissertation]. Graduate Department of Electrical and Computer Engineering, University of Toronto; 2009.
[12] Dueck D, Frey B. Non-metric affinity propagation for unsupervised image categorization. In: IEEE International Conference on Computer Vision (ICCV). 2007. p. 1–8.
[13] Frey BJ, Dueck D. Clustering by passing messages between data points. Science 2007;315(5814):972–6.
[14] Givoni IE. Beyond affinity propagation: message passing algorithms for clustering [dissertation]. Graduate Department of Computer Science, University of Toronto; 2012.
[15] Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Computing Surveys 1999;31:264–323.
[16] Abraham A, Das S, Roy S. Swarm intelligence algorithms for data clustering. In: Maimon O, Rokach L, editors. Soft Computing for Knowledge Discovery and Data Mining. Germany: Springer Verlag; 2007. p. 279–313.
[17] Haddadi H, King AJ, Wills AP, Fay D, Lowe J, Morton AJ, et al. Determining association networks in social animals: choosing spatial–temporal criteria and sampling rates. Behavioral Ecology and Sociobiology 2011;65(8):1659–68.
[18] Givoni IE, Frey BJ. A binary variable model for affinity propagation. Neural Computation 2009;21(6):1589–600.
[19] He Y, Chen Q, Wang X, Xu R, Bai X, Meng X. An adaptive affinity propagation document clustering. In: 2010 The 7th International Conference on Informatics and Systems (INFOS). 2010. p. 1–7.
[20] Quera V, Beltran FS, Dolado R. Flocking behaviour: agent-based simulation and hierarchical leadership. Journal of Artificial Societies and Social Simulation 2010;13(2) [about 19 pp.] Available from: http://jasss.soc.surrey.ac.uk/13/2/8.html
[21] Zhang L, Du Z. Affinity propagation clustering with geodesic distances. Journal of Computational Information Systems 2010;6(1):47–53.
[22] Huang Y, Huang K, Tan T. Computational primitives of visual perception. In: In: 2009 16th IEEE International Conference on Image Processing (ICIP). 2009. p. 1793–6.
[23] Prim C. Shortest connection networks and some generalizations. The Bell System Technical Journal 1957;36:1389–401.
[24] Floyd RW. Algorithm 97. Shortest path. Communications of the ACM 1962;5(6):345.
[25] Leone M, Sumedha, Weigt M. Unsupervised and semi-supervised clustering by message passing: soft-constraint affinity propagation. European Physical Journal B 2007;66(1).
[26] Reeder CC. A novel computational method for inferring dynamic genetic regulatory trajectories [dissertation]. Massachusetts Institute of Technology; 2008.
[27] Yang J, Wang Y, Somya A, Xiu J, Li Z, Zhang B. Spatial–temporal affinity propagation for feature clustering with application to traffic video analysis. In: Kimmel R, Klette R, Sugimoto A, editors. Lecture Notes in Computer Science. ACCV'10 Proceedings of the 10th Asian conference on Computer Vision, Part II. vol. 6493. Springer; 2011. p. 606–18.
[28] Kschischang FR, Frey BJ, Loeliger H. Factor graphs and the sum–product algorithm. IEEE Transactions on Information Theory 2001;47(2):498–519.
[29] Huth A, Wissel C. The simulation of fish schools in comparison with experimental data. Ecological Modelling 1994;75–76:135–45.
[30] Bodenhofer U, Kothmeier A, Hochreiter S. APCluster: an R package for affinity propagation clustering. Bioinformatics 2011;27:2463–4.
[31] Hamming RW. Error detecting and error correcting codes. The Bell System Technical Journal 1950;29(2):147–60.
[32] Rand WM. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 1971;66(336):846–50.
[33] Okabe A. Statistical analysis of the pattern similarity between two sets of regional clusters. Environment and Planning A 1981;13(5):547–62.
[34] Hubert L, Arabie P. Comparing partitions. Journal of Classification 1985;2:193–218.
[35] Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. Journal of Machine Learning Research 2010;11:2837–54.
[36] Meilă M. Comparing clusterings – an information based distance. Journal of Multivariate Analysis 2007;98:873–95.
[37] Steinley D. Properties of the Hubert–Arabie Adjusted Rand Index. Psychological Methods 2004;9(3):386–96.

[38] Meilă M. Local equivalences of distances between clusterings – a geometric perspective. Machine Learning 2011;86(3):369–89.
[39] Santos JM, Embrechts M. On the use of the Adjusted Rand Index as a metric for evaluating supervised classification. In: In: ICANN'09, Proceedings of the 19th International Conference on Artificial Neural Networks: Part II. Springer; 2009. p. 175–84.
[40] Halkidi M, Batistakis Y, Vazirgiannis M. On clustering validation techniques. Journal of Intelligent Information Systems 2001;17(2–3):107–45.
[41] Segment (Version 1.1). Barcelona: Econceptes; 2012 [Computer software].
[42] Quera V, Herrando S, Beltran FS, Salas L, Miñano M. An index for quantifying flocking behavior. Perceptual and Motor Skills 2007;105:977–87.