Analysis and Classification of Collective Behavior Using Generative Modeling and Nonlinear Manifold Learning

Sachit Butail[a], Erik M. Bollt[b], Maurizio Porfiri[a,*]

[a]*Department of Mechanical and Aerospace Engineering, Polytechnic Institute of New York University, Brooklyn, NY 11201, USA*
[b]*Department of Mathematics and Computer Science, Clarkson University, Potsdam, NY 13699, USA*

Abstract

In this paper, we build a framework for the analysis and classification of collective behavior using methods from generative modeling and nonlinear manifold learning. We represent an animal group with a set of finite-sized particles and vary known features of the group structure and motion via a class of generative models to position each particle on a two-dimensional plane. Particle positions are then mapped onto training images that are processed to emphasize the features of interest and match attainable far-field videos of real animal groups. The training images serve as templates of recognizable patterns of collective behavior and are compactly represented in a low-dimensional space called embedding manifold. Two mappings from the manifold are derived: the manifold-to-image mapping serves to reconstruct new and unseen images of the group and the manifold-to-feature mapping allows frame-by-frame classification of raw video. We validate the combined framework on datasets of growing level of complexity. Specifically, we classify artificial images from the generative model, interacting self-propelled particle model, and raw overhead videos of schooling fish obtained from the literature.

Keywords: classification, collective motion, fish schooling, generative modeling, Isomap

1. Introduction

The study of collective behavior is often complemented with the analysis of high-volume datasets available in the form of simulated trajectories [1, 2, 3] and videos [4]. Patterns in these datasets are recognizable to a trained observer, who can quickly determine whether a set of particles or a school of fish is moving together in coordination or in complete disorder. However, this standard of recognition is not available at a machine level, where we must classify the trajectory data by fitting it into activity models [5, 6, 7]. The intermediate process of multi-target tracking is a computational overhead that scales with the number of animals observed [8, 9]. Although a naive comparison of images to an exhaustive database of training videos would preclude the need to track individuals, it would shift the computational burden from processing to storage. Instead, an enabling requirement for a fast, data-driven approach is to store recognizable patterns of collective behavior in a compact representation so that they can be archived and retrieved for quick comparison.

Assuming no coordination among individuals, we expect the trajectories of group members to be independent of each other, thus requiring a large number of degrees of freedom to describe the group motion; trajectories of coordinated individuals should instead be manifested through fewer degrees of freedom related to the movement of select group members. This realization forms the first step to reducing the data to a few important features that can faithfully represent the ongoing process. For example, images of animal groups may be classified on the basis of features of the spatial distribution, such as number of subgroups, population density, and group configuration, and features of the dynamics, such as change in group size, orientation, and speed. As a group of animals maneuver through space, temporal variation of these features can inform the nature of the interaction between them [10, 11].

*Corresponding author
Email addresses: sb4304@nyu.edu (Sachit Butail), mporfiri@poly.edu (Maurizio Porfiri), bolltem@clarkson.edu (Erik M. Bollt)

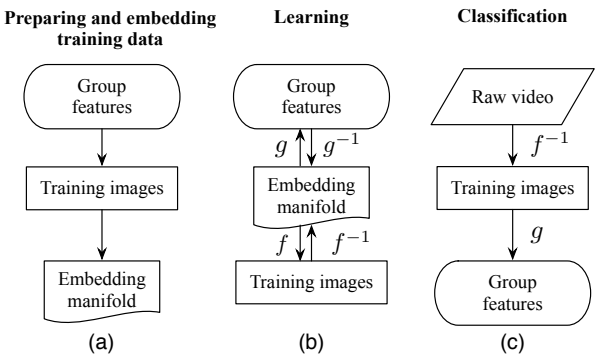

Figure 1: (a) We use generative models to create synthetic training images that match a far-field view of animal groups. The training images are then compactly represented by the Isomap algorithm to a reduced dimension form called the embedding manifold. (b) We then derive invertible functions from the embedding manifold to the training images as well as group features. (c) The mappings are subsequently used to classify raw images of collective behavior.

Dimensionality reduction is the process of identifying low-dimensional representations that preserves the dissimilarities between points on a high-dimensional space [12, 13]. The high-dimensional data may be in the form of positions of multiple individuals or even raw images. Once generated, these representations, called embedding manifolds, can be used in a class of machine learning algorithms, called manifold learning, for analysis and classification. A similar approach is successfully implemented in human activity recognition [14, 15], face recognition [16], pose estimation [14, 17], exploration of video sequences [18], and handwriting recognition [19]. If the difference between two points is accurately represented by the Euclidean distance between them [20], linear dimensionality reduction methods, such as principal components analysis and singular value decomposition can be used. These methods transform the dataset along directions of maximum variability, thereby preserving relative configuration. If the points on the input data cannot be faithfully differentiated by the Euclidean distance, nonlinear methods such as isometric mapping (Isomap) [19] and local linear embedding (LLE) [21, 22, 23] may be used. As an example, the distance between two cities on earth is correctly represented along the great circle (geodesic) and not a straight line [24]. Similarly, images of human faces or handwritten letters are not usefully separated by a linear difference of intensity values [19]. In earlier work, we have used Isomap to show that dimensionality of the low-dimensional embedding created from images of self-propelled particles is indicative of the degree of coordination [25] as well as the number of subgroups [26]. In contrast to the frame-by-frame classification method that is developed in this paper, inferences from the dimensionality of the embedding manifold in [25, 26] are made on the basis of long sequences, comprising a few thousand frames.

The application of manifold learning for analysis and classification begins with the collection of training data for sampling the input space. The success of manifold learning in image-based analysis depends on several factors, including the variability and number of training images [27] and image representation [18, 28]. In face recognition, for example, a large number of centered images of the subject are used to exhaustively sample the expected embedding space [19, 21]. Similarly, for pose estimation, multiple viewpoints and extended videos of human activity are used as training data [14]. Whereas humans can be requested to move in a specific manner, achieving a similar task with real animals is impractical. Consequently, individual frames are tagged by experts to quantify specific behaviors followed by classification on the basis of goodness of fits [29].

We investigate the possibility of using generative models to create training images of collective behavior that can then be used to classify real videos. A generative model is a probabilistic relation mapping features to observations; for example, the probability of an image of an animal group given the number of subgroups. Once a generative model is available, it may also be inverted so that features can be extracted from observations [30, 31]. To create training images, we generate particle positions that emphasize variations of group features and project them on a two-dimensional image plane. In this data-driven approach, we do not propose a new model of collective behavior, rather a method to generate images of group configurations and trajectories that efficiently sample an underlying feature space. We use the Isomap algorithm to embed these images on a low-dimensional manifold (Fig. 1a). The Isomap algorithm approximates the embedding manifold in the higher dimensional space by first constructing geodesics between two points. It then uses classical multidimensional scaling to construct a low-dimensional representation between points to

generate an isometric embedding. That is, distances within the manifold infinitesimally respect the Euclidean distances [32]. (More details on the Isomap algorithm are given in Section 3.) Next, we learn the manifolds by deriving two invertible mappings: one from the manifold to the input images and the other from the manifold to the underlying feature space (Fig. 1b). These mappings are used to classify new unseen images of animal groups (Fig. 1c).

We test the mappings on datasets with growing level of complexity. First, the manifold-to-image mapping is verified by reconstructing new images after interpolation on the manifold. Next, new images created using generative models are classified using the manifold-to-feature mapping. We then analyze and classify images of self-propelled particles interacting via a modified Vicsek model [33], where the particles randomly change their speed to simulate speed variations observed in animal groups [34, 35]. Finally, the approach is validated on experimental videos of zebrafish schooling in laboratory controlled environments obtained from the literature [36]. The contributions of this paper are

i) We present generative models to build training images training images of a set of finite-sized particles that emphasize features of collective motion;

ii) We embed the dataset of training images onto a low-dimensional manifold representation using nonlinear dimensionality reduction, and build invertible mappings from the low-dimensional manifold to the training images and the group features;

iii) We classify unused training images and images of self-propelled particles based on group structure and motion; and

iv) We demonstrate the proposed framework to classify raw videos of schooling zebrafish based on group structure.

This paper is organized as follows. In Section 2, we present generative models for a group of finite-sized particles to create training images of collective behavior. In Section 3, we briefly describe the Isomap algorithm and implement it on the training images to create reduced dimensional manifolds and associated mappings. Section 4 evaluates the manifold mappings through interpolation on the manifold to create new meaningful images in the feature space followed by classification of artificial datasets. Section 5 illustrates a possible application of the methods developed in this paper to analyze real images of schooling zebrafish. We conclude in Section 6 with a discussion of the performance of the manifold learning approach including limitations that are being addressed in ongoing work. Appendix A gives the mathematical description of the generative model for group structure used in this paper. Appendix B describes the procedure for learning the invertible mappings from the manifold to images and features.

2. Generative models of collective behavior

We identify the following group features to differentiate one group from another, or the same group at different times:

i) *Group structure* characterizes the spatial distribution of members within a group. A group may consist of one or more subgroups of different size. A subgroup can be usefully defined as a spatially non-overlapping cluster of group members, whose nearest-neighbor belong to the same subgroup.

ii) *Group motion* characterizes the movement of the group members. A group may move with high or low degree of coordination with different average speeds.

Notation-wise we denote the number of members in a group by N. Index i or j, wherever specified, is used to identify a single member or a subgroup. With reference to a Cartesian coordinate system, the state of the i-th member moving in a plane is described in terms of the position, velocity, and orientation, which are denoted by $\boldsymbol{r}_i$, $\boldsymbol{v}_i$, and θ_i respectively (vectors, wherever needed, are displayed in bold font). Discrete time-step is denoted by index k or l. The length of a time-step, Δt, is kept the same for generating training images as well as test images. Group members simulated as finite-sized particles move in a square domain of side L. The size of each particle is a sphere of radius $L/100$.

2.1. Generative model

Generative models describe a probabilistic relationship between observations of a process and and features of interest that are typically unobservable [37]. Examples of observations are images, positions, and trajectories of individuals in a group. Examples of features are number of subgroups and degree of coordination between members. We use generative models to modify the group features of a set of particles and project the resulting trajectories on a virtual camera plane to create synthetic images of collective behavior. Individual features are varied to exhaustively sample the input space and the variation is scaled to generate the points sequentially.

More formally, given a scaling $\nu \in [0, 1]$, we generate an N-member particle group whose properties vary according to features γ_1 and γ_2 that are functions of ν. We use a spiral function of ν to vary each feature gradually as we explore the underlying space. (As we note later in Section 3, a spiral form allows an easy way to verify the correct embedding manifold.) An n loop spiral centered at p_{10}, p_{20} with range $2p_1, 2p_2$ is given by

$$\begin{aligned} \gamma_1(\nu, p_1) &= p_1 \nu \cos(2\pi n\nu) + p_{10} \\ \gamma_2(\nu, p_2) &= p_2 \nu \sin(2\pi n\nu) + p_{20}. \end{aligned} \tag{1}$$

The two-dimensional position $\boldsymbol{r}_i$ of the i-th member is generated according to a probability distribution function formulated in the form of a Gaussian noise model as $P(\boldsymbol{r}_i|\gamma_1, \gamma_2) = \mathbb{N}(\boldsymbol{r}_i, f(\gamma_1, \gamma_2), \sigma), i = 1, \ldots, N$, where $\mathbb{N}(\boldsymbol{r}_i, f(\gamma_1, \gamma_2), \sigma)$ is the Normal distribution function evaluated at $\boldsymbol{r}_i$ with mean $f(\gamma_1, \gamma_2)$ and standard deviation σ. In what follows, we prefix the features as ${}^S\gamma_1$ and ${}^S\gamma_2$ or ${}^M\gamma_1$ and ${}^M\gamma_2$ for group structure or motion.

2.1.1. Group structure

Group structure represents the spatial distribution of individual members within the group. Specifically, a group may be clustered, in which case we observe multiple subgroups or uniformly distributed, in which case no distinguishable subgroups may be found. In addition, the size of each subgroup may vary. The formation of subgroups in animal groups can be triggered due to diverging leaders [38, 39], multiple foraging sites [40], and predator attacks that cause a large group to split [41]. In self-propelled particle models of collective behavior, subgroups arise in conditions where there is high coordination in a sparsely distributed particle set [42, 43, 44, 45].

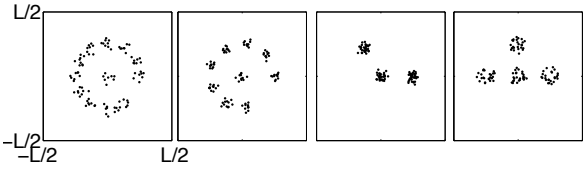

Figure 2: Selected instances of a hundred-particle set generated using Eqn. (2) on a square domain of side L. The arrangement of particles varies in the number of subgroups and the size of each subgroup.

To generate data for group structure, we sample individual particle positions into subgroups within the observation region. Subgroups are spatially isolated from each other to visually highlight the difference between different number of subgroups. The size and shape of the observation region may be used to determine the arrangement of the subgroups. Group structure is varied on the basis of the number of subgroups (${}^S\gamma_1$) and the size of each subgroup (${}^S\gamma_2$); the position of i-th member $\boldsymbol{r}_i$ is

$$\boldsymbol{r}_i = f({}^S\gamma_1, {}^S\gamma_2) + \boldsymbol{\xi}_s, \tag{2}$$

where the level of confidence in the model is denoted by the noise $\boldsymbol{\xi}_s$ sampled from a zero-mean Gaussian distribution with standard deviation $\boldsymbol{\sigma}_s$. The function $f({}^S\gamma_1, {}^S\gamma_2)$ is a geometric mapping that takes as input the number of subgroups and size and outputs particle positions. Figure 2 shows four instances of group structure where subgroups are located on a circle including its center (equations in Appendix A) with $N = 100$ and $\boldsymbol{\sigma}_s = [L/60, L/60]^{\mathrm{T}}$, where T denotes matrix transpose. Note that this is one way to generate subgroups from a fixed number of particles. Alternatively, models inspired from foraging [46] or movement in bounded space [47] may be used, however, being stochastic in the dynamics, these models do not give full control over the parameters of interest required for producing a predictable manifold embedding.

2.1.2. Group motion

Group motion represents movement within a group. A group may be coordinated, in which case all members move in one direction, or uncoordinated, in which case the movement is random. Similarly, the average speed of the group may be high or low. We use a generative model to create short trajectories that capture variation in group polarization, defined as the degree of alignment of the group [33], and group speed. Classification using these features alone can be used to detect actions of members such as i) moving away from each other, ii) moving towards each other, and iii) freezing. In addition, a combination of these actions can be used to detect behaviors such as foraging [48], which may be characterized as a movement towards a source; obstacle avoidance [49], which may be characterized as sudden change in polarization with possible increase in the number of subgroups; and escape from a predator [50], which may be characterized as a sudden change in polarization and speed with possible increase in the number of subgroups.

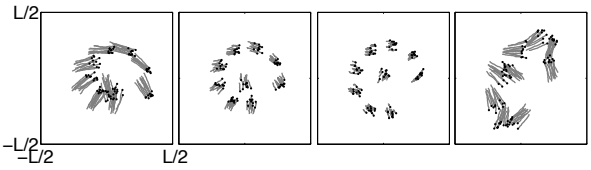

Figure 3: Selected instances of ahead of motion tracks of a hundred-particle set from Eqn. (3) on a square domain of side L. The motion of particles varies in the range of orientations and average speed.

Position information of each member alone may not be sufficient to infer he characteristics of group motion. Therefore, we sample motion traces of constant velocity particles whose initial direction of motion and speed is varied. Specifically, given a group of N members, the group motion is varied on the basis of the speed of the particles (${}^{M}\gamma_1$) and the range of orientations (${}^{M}\gamma_2$); a motion trace, $\boldsymbol{r}_i[k], k = 0, \ldots, T$, of the i-th member is generated as

$$\boldsymbol{r}_i[k] = \boldsymbol{r}_i[k-1] + {}^{M}\gamma_1 \Delta t \begin{bmatrix} \cos(\vartheta_i) \\ \sin(\vartheta_i) \end{bmatrix} + \boldsymbol{\xi}_m, \tag{3}$$

where $\vartheta_i = -{}^{M}\gamma_2(2i - N)/N$, and $\boldsymbol{\xi}_m$ is the noise sampled from a zero-mean Gaussian distribution with standard deviation $\boldsymbol{\sigma}_m$ denoting the level of confidence in the model. Figure 3 shows four instances of group motion with $N = 100, T = 36, \Delta t = 1$ and $\boldsymbol{\sigma}_m = [L/1000, L/1000]^{\mathrm{T}}$. Generating group motion using (3) does not change the average direction of the motion, implying for example that a coordinated group will always move left to right. This rotational dependence occurs at the level of both the group structure and motion images, and can be addressed by adding average group direction as a third feature to the list of features. As a stepping stone for ascertaining the validity of the method, we focus on group speed and group coordination in this paper.

2.2. Image representation

We seek final observations in the form of images of animal groups filmed from a far-field view. To obtain an image where the particle size matches the animals we expect to film, we model each particle as a sphere and project it orthographically onto an image plane [51]. (Note that we approximate far-field view as orthographic projection although a perspective projection model may also be used.) The resulting $p \times p$ pixel foreground image at time k, $F[k] \in \mathbb{R}^{p \times p}$, is a binary occupancy grid with non-zero values indicating the presence of a group member. Because of partial occlusions, a binary foreground may not accurately compare instances of different group densities. Similarly, within-group movement may be suppressed due to translational movement of the group as a whole. We therefore process the foreground image $F[k]$ to emphasize the distribution and movement, respectively.

For emphasizing group structure, our goal is to exaggerate relative location of each particle to discriminate a densely spaced subgroup from a sparsely spaced one. To exaggerate particle position so that each particle appears as an extended object we blur the foreground by convolving it with a Gaussian kernel [52]. The blurred foreground F' is

$$F'[k] = F[k] * G_b, \tag{4}$$

where G_b is a two-dimensional Gaussian kernel and $*$ indicates the convolution integral. The size and standard deviation of the kernel in pixels are selected on the basis of the size of a group member relative to the image. We

set the value of $\boldsymbol{\sigma}_b$ equal to the size of the observation region divided by the number of particles. Therefore, for N particles in a square of side L, we set $\boldsymbol{\sigma}_b = [L/N, L/N]^{\mathrm{T}}$. (The window size of the kernel is also set at $[L/N, L/N]^{\mathrm{T}}$). Finally, we make the image of the particle set translation invariant by subtracting the centroid of the non-zero pixels in the blurred foreground from each pixel.

For group motion, in order to compare movement within a sequence of images, our goal is to encode spatio-temporal information compactly in a single motion history image. A motion history image (MHI) is created by adding successive images whose intensity is weighted by the difference in time from the current image. Given the current time-step k, and a cut-off value T, $H[k]$, is created recursively as [53]

$$H_{u,v}[k] = \begin{cases} T & \text{if } F_{u,v}[k] = 1 \\ \max(0, H_{u,v}[k-1] - 1) & \text{otherwise,} \end{cases} \tag{5}$$

where $F_{u,v}[k]$ is the value of the foreground pixel at v-th row and u-th column in the k-th frame, and T determines the number of frames that form the MHI. MHIs of group motion consist of straight lines for a smaller speeds, and long lines for higher. Similar to group structure images, we blur and center each MHI by first convolving it with the same Gaussian kernel as in (4) and then subtracting the centroid of foreground pixels.

3. Using Isomap to construct manifolds of collective behavior

In this section, we describe the methodology to construct low-dimensional representations of collective behavior from the training images. We use the nonlinear dimensionality reduction algorithm called Isomap to build a reduced-dimension representation of the training images. The input to the Isomap algorithm is a set of points in d-dimensional space, a distance metric on these points, and a neighborhood defining parameter. The output is a corresponding set of points in an e-dimensional embedding manifold such that $e << d$.

3.1. Isomap on training images of collective behavior

The Isomap algorithm begins by constructing a neighborhood graph based on the distance between points that are κ-nearest neighbors or within an ϵ distance. The resulting distance matrix consisting of pairwise distance between all points is then updated by computing geodesics along the manifold. Finally, the e-dimensional manifold is obtained by performing classical multidimensional scaling (MDS) on the updated distance matrix [32]. Classical multidimensional-scaling preserves the pairwise distances between points on the high-dimensional manifold at a lower dimension by minimizing a cost function that penalizes the difference in distances between pairs of points in each dimension [32]. Candidate embeddings are evaluated on the basis of residual variances in the reconstruction error. The dimensionality of the embedding manifold is typically noted by an elbow in the plot of residual variances against dimensionality [19], denoting little improvement in the reconstruction accuracy beyond a given value.

The value of d for images is equal to the number of pixels. (An image with resolution 100×100 pixels has $d = 10000$.) The neighborhood defining parameter that we use is based on κ nearest neighbors. The distance metric $d(k, l)$ that we use to capture the dissimilarity between two foreground images $F'[k]$ and $F'[l]$ is

$$d(k, l) = \|F'[k] - F'[l]\|, \tag{6}$$

where $\|\cdot\|$ is the Euclidean norm of the vectorized image intensity matrix in $\mathbb{R}^{p \times p}$. (We use the same notation for Euclidean norm in $\mathbb{R}^p$.)

To learn a manifold of collective behavior, we seek i) a fully connected graph that projects all images on the low-dimensional space, and ii) minimal illegal edges called shortcuts that connect two points which are not nearest neighbors. Varying input group features along a spiral in feature space provides a quick way to identify the right low-dimensional manifold. This is because a spiral should be a one-dimensional curve and shortcuts are easily identified as two points that are connected but not sequential.

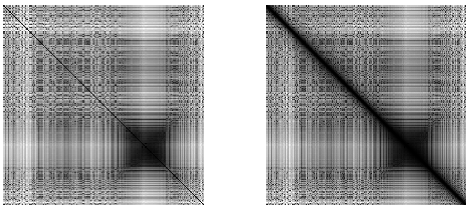

Figure 4: (a) Original and (b) weighted distance matrix between training images after it is temporally weighted. A point on each axis represents a training image. Darker regions imply proximity in distance. The final distance matrix brings sequential images closer together.

3.2. Temporal weighting

A straight-forward implementation of the Isomap algorithm is unable to extract the true variation in training images for a range of κ values and the number of training images. This can be attributed to several reasons including the stochasticity of data, a distance function that does not accurately represent dissimilarity in images, and rounding off due to pixel representation.

Since we generate images that are sequentially close to each other on the desired feature space, we include an additional step to weight values in the distance matrix so that images that are close to each other in the feature space are close in the image space as well. Specifically, for the foreground image at time k, the weighted distance $d'(k,l)$ between the k-th and l-th frame is

$$d'(k,l) = d(k,l)\exp\left(\frac{-\lambda}{|k-l|}\right), \tag{7}$$

where λ is a tuning parameter that emphasizes low-dimensionality as its value is increased. A straightforward implementation ($\lambda = 0$) of the Isomap algorithm on the training images consistently results in a high-dimensional manifold. We therefore increase the value of λ until a two-dimensional manifold is obtained. Figure 4 shows a distance matrix before and after the temporal weighting step is applied to a 100×100 distance matrix with $\lambda = 2$. In our tests, we find that the values of κ-nearest neighbors and λ that yield the desired two-dimensional manifold are in proportion to the number of training images. (Although the value of λ can be increased to obtain a completely one-dimensional manifold, it also leads to an increase in shortcuts.)

3.3. Group structure manifold

The group structure manifold highlights variation along the number of subgroups and the size of each subgroup. Figure 5a shows the input feature curve for generating training images of sixteen finite-sized particles according to (2). We set $p_{10} = 3 \times 10^{-2}L, p_1 = 2 \times 10^{-2}L$ (size of the subgroup), and $p_{20} = N/2, p_2 = N/2$ (the number of subgroups). Figure 5b shows the resulting manifold in the form of a neighborhood graph after running Isomap on three hundred training images ($\kappa = 4$, $\lambda = 2$). The neighborhood graph has a few shortcuts between nodes that are otherwise non-adjacent in the input space. To ensure consistency in the choice of parameters, we rerun the Isomap algorithm multiple times on a randomly chosen ninety percent subset of training images and ascertain that the dimensionality stays the same.

3.4. Group motion manifold

The group motion manifold embeds MHIs of particles whose speed and orientations vary. Since the MHI depends on the particle positions in the first frame, each point on the group structure manifold gives rise to an initial position for generating training images. Given a centered image of a group structure we generate trajectories according to (3). In this example, we set $p_{10} = 1/3 \times 10^{-2}L/\Delta t, p_1 = 1/3 \times 10^{-2}L/\Delta t$ (average speed), and $p_{20} = \pi/2, p_2 = \pi$ (range of orientations). MHIs are generated using $T = 36\Delta t$ to obtain distinct traces of particle motion on the image. The resulting MHIs resemble faded streaks with varying intensity whose length and direction depend on the individual particle speed and direction of motion. The values of $\kappa = 3$ and $\lambda = 1$ for a hundred MHIs. (As before for group structure, we verify the choice of κ and λ by running the Isomap algorithm multiple times on a subset of training set.) Figure 6 shows the input feature curve with the resulting manifold in the form of a neighborhood graph.

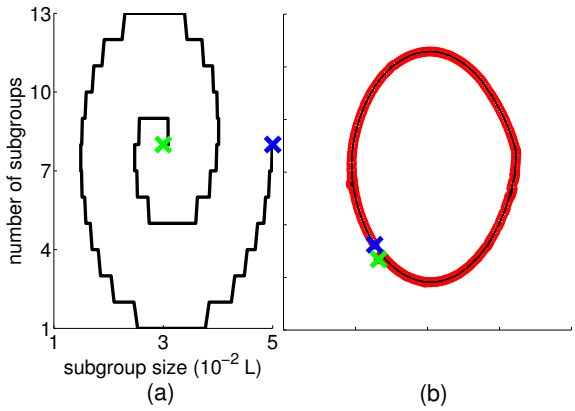

Figure 5: Group structure manifold. (a) The training set comprises images of a set of sixteen particles which are arranged according to variation in number of subgroups and the size of each subgroup. (b) Neighborhood graph on the resulting manifold. The start and end points are marked using a green and blue cross respectively.

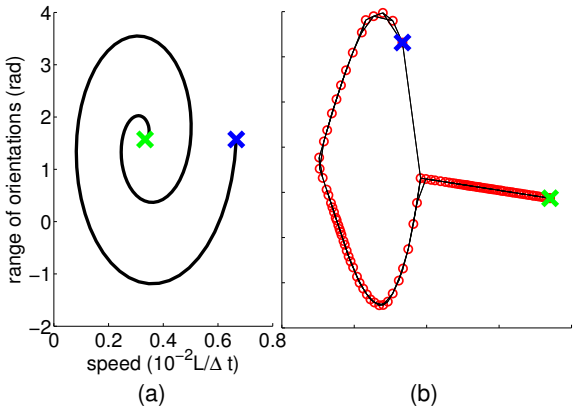

Figure 6: Group motion manifold. (a) The training set comprises images of a set of sixteen particles which move according to variation in combined velocity. (b) Neighborhood graph on the resulting manifold. The start and end points are marked using a green and a blue cross respectively.

3.5. Manifold-to-image and manifold-to-feature mapping

Once built, two mappings are learnt for each manifold: the manifold-to-image mapping, $f : \mathbb{R}^e \rightarrow \mathbb{R}^d$ is used to create images similar to the training images from a point on the manifold, and the manifold-to-feature mapping $g : \mathbb{R}^e \rightarrow \mathbb{R}^2$ is used to classify a point on the manifold based on the group features. The method of deriving each mapping is the same and is detailed in Appendix B.

The mappings f and g is obtained in the form of a matrix. In the case of manifold-to-image mapping, the inverse mapping f^{-1} is used to project test images onto points on the manifold space. These points are then projected onto the feature space by using the manifold-to-feature mapping g (Fig. 1c). Similar to the notation used for features, we prefix the mapping for a given manifold as ${}^S f, {}^S g$ for group structure, and ${}^M f, {}^M g$ for group motion.

4. Analysis and classification

In this section, we validate the manifold-to-image and manifold-to-feature mappings on a series of synthesized datasets. We first use the manifold-to-image mapping to interpolate between alternate images in the training dataset; the newly reconstructed images are compared to real unused images. Next, the manifold-to-feature mapping is verified by projecting unseen test images on the manifold. The recovered features are compared with the available ground truth data in the form of known features. Finally, we classify images of finite-sized interacting self-propelled particles.

4.1. Interpolation on the manifold

A first step in the verification of manifold-to-image mapping can be performed by interpolating on the manifold to produce new images. A nonlinear interpolation (as opposed to linear interpolation) on the image space generates images that vary along the group features (as opposed to pixel intensities). Nonlinear interpolation has application in enriching real videos recorded at a low frame rate [54], a situation that may occur due to low lighting or limited storage.

To interpolate on the manifold, we use alternate images from training sequence to create the manifold, leaving every other image unused. Successive two-dimensional points on this manifold subset are linearly interpolated and new images are constructed using the manifold-to-image mapping. Figures 7a and b show a select set of interpolated images for group structure and group motion. The new images after interpolation on the manifold follow the same trend as the input images. The performance of this method is quantified by computing a Euclidean norm between the vectorized interpolated image and the unused image. We refer to this value as pixel-placement error, since a mismatched pixel in either of the 100×100 pixel image will increase the error by one. For group structure, the pixel-placement error for the full training sequence is 7.9 ± 3.2 pixels, whereas for group motion the pixel-placement error is 5.1 ± 4.6 pixels. In contrast the direct linear interpolation between images yields an average pixel-placement error of 15.2 ± 1.6 pixels for group structure and 23.8 ± 2.7 pixels for group motion.

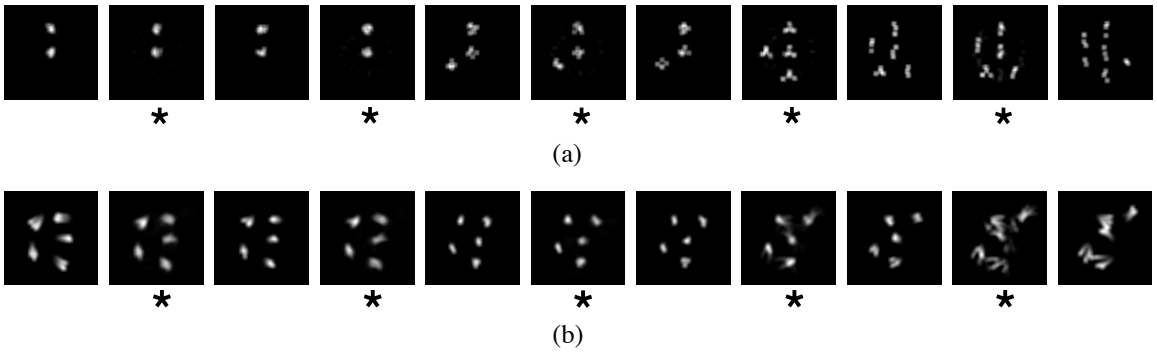

Figure 7: Training images are interpolated on the (a) group structure and (b) group motion manifold to produce new images marked with a (*).

4.2. Classification of test images

We create test images from generative models described in Section 2 to determine if previously unseen images which appear similar to training images can be classified correctly. The value of input features to the generative model serve as ground-truth for comparison. We first use the inverse of manifold-to-image mapping to obtain a two-dimensional point on the reduced-dimension space. This point is then mapped onto the feature space by using the manifold-to-feature mapping for classification (Fig. 1c).

Figure 8a compares the value of the recovered subgroup size and number of subgroups in the test images to the true values. Since the test images are generated sequentially, we smooth the recovered features with a moving average of window size five. We observe less error in the number of subgroups than in the subgroup size. This can be attributed to the dependence of the group structure on the size of the observation region, where a large subgroup may be perceived as multiple subgroups.

For group motion, we project test MHIs of the same initial group structure onto the two-dimensional manifold space of group motion. The resulting points are then classified into values of speed and range of orientations. Figure 9a compares these values to known features (ground-truth). Compared to group structure, classification of group motion is more accurate, possibly due to smoother variation in images as the group motion features are varied. The test images are projected close to the manifold as shown in Fig. 9b.

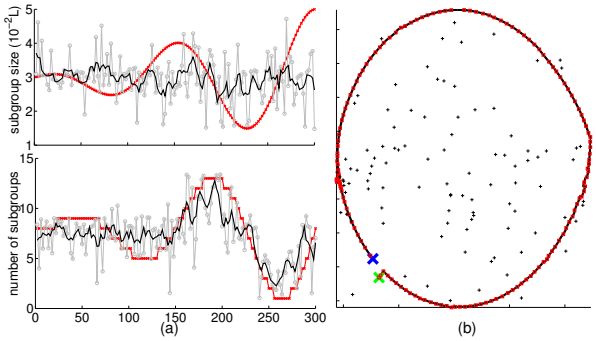

Figure 8: Recovering the number of subgroups and subgroup size from test images generated using the model Eqn. (2) and projected on the manifold. (a) The ground truth (red) is compared to the projected values (grey) and smoothed values (black). (b) Direct projection of test images on the manifold. Red squares denote the centers of the radial basis functions.

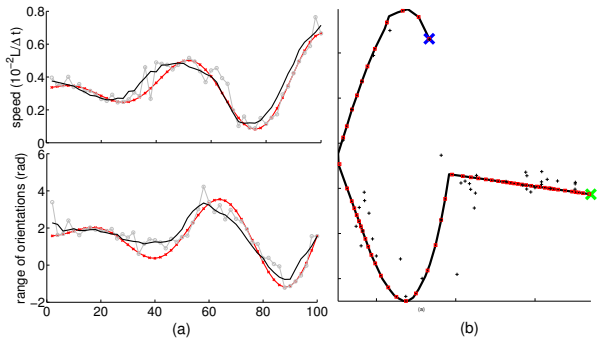

Figure 9: Recovering group speed and polarization from test images generated using the model Eqn. (3) and projected on the manifold. (a) The ground truth (red) is compared to the projected values (grey) and smoothed values (black). (b) The projections on the manifold. Red squares on the manifold denote centers of radial basis functions.

4.3. Self-propelled particle model with Poisson process change in speed and orientation

To validate the framework on a synthetic dataset representative of collective motion, we simulate five hundred frames of $N = 16$ self-propelled finite-sized particles interacting according to the Vicsek model [33]. The Vicsek model updates a particle's position as a function of its nearest neighbors. Given the position $\boldsymbol{r}_i$ of the i-th particle, we define its nearest neighbors $\mathcal{N}_i$ as the set of particles that are within a metric distance ($\|\boldsymbol{r}_i - \boldsymbol{r}_j\| < r_d$), including the i-th particle. The orientation $\theta_i[k]$ and position $\boldsymbol{r}_i[k]$ are updated as [33]

$$\begin{aligned} \theta_i[k] &= \arg(\hat{\boldsymbol{v}}_i[k]) + \Delta\theta, \\ \boldsymbol{r}_i[k+1] &= \boldsymbol{r}_i[k] + s_i[k] \begin{bmatrix} \cos(\theta_i[k+1]) \\ \sin(\theta_i[k+1]) \end{bmatrix} \Delta t, \text{ where} \\ \hat{\boldsymbol{v}}_i[k] &= \frac{1}{|\mathcal{N}_i[k]|} \sum_{j \in \mathcal{N}_i[k]} \begin{bmatrix} \cos(\theta_j[k]) \\ \sin(\theta_j[k]) \end{bmatrix}, \end{aligned} \tag{8}$$

$s_i[k]$ is the speed, and $\Delta\theta$ is noise sampled from a uniform distribution with interval $[-\eta/2, \eta/2]$. The value of the parameter η determines the degree of coordination between nearest neighbors (a high value results in low coordination). The initial position of the particle set $\boldsymbol{r}_i[0], i = 1, \ldots, N$ is generated using the group structure generative model (2); the initial orientation is set to zero. The simulations are carried out in the square domain of size L with a periodic boundary; a particle that crosses a boundary edge emerges from the edge directly opposite to it. In the original Vicsek model, the speed $s_i[k]$ of all particles is constant. We modify the motion update (8) to incorporate sudden changes in speed of the particles according to a Poisson process [55]. Specifically, the probability of a change in speed at time k is $p = \exp(-\Lambda)\Lambda$, where Λ is the rate parameter of the Poisson process. The value of Λ is a function of Δt and

determines the frequency of occurrence of an event within a given time. At each time step a uniformly random variable U is sampled on the interval $[0, 1]$. If $U \leq p$, the speed of all the particles is updated as

$$s_i[k+1] = s_i[k] + \xi_p,\ i = 1, \ldots, N, \tag{9}$$

where ξ_p is sampled from zero-mean Gaussian distribution with standard deviation σ_p. We set the value of $\Lambda = \Delta t/3$, and σ_p equal to one-tenth the initial average speed. Six simulations spanning a range of values for each parameter $\eta = \{0.005, 0.05, 0.5\}$ and $r_d = \{L/80, L/40\}$ are run for five hundred time steps. As with the training images, each particle is modeled as a finite sized sphere and projected onto the image plane to obtain a binary foreground frame. Blurred and centered foreground images are projected onto the group structure manifold. Similarly blurred and centered motion history images are projected on the corresponding group motion manifold.

Other mathematical models that describe collective motion in terms of self-propelled particles include [44] and [45]. In [44], an individual's motion is determined by behavioral rules based on spherical interaction zones around its position; the presence of another particle in the zones determine if two particles will attract, align, or repel each other. In [45], the orientation of an individual particle is updated on the basis of a weighted values of interaction force that tends to align the motion of two particles and radial force that brings them together. Similar to the Vicsek model [33], a set of parameters within each of these can be tuned to display emergent behaviors representing collective motion. We use the Vicsek model as it offers a simple set of tuning parameters to span a broad spectrum of collective motions.

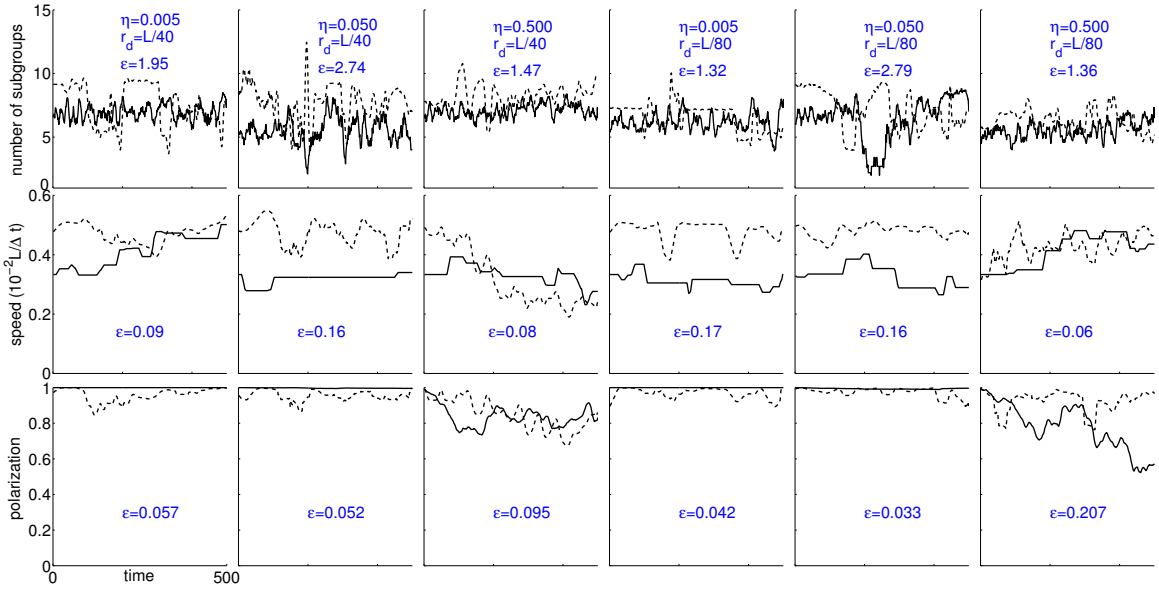

Figure 10: Classifying six different simulations of self-propelled particles on the basis of number of subgroups, average speed, and polarization of the particles. In each case, the estimated value (dashed) is compared with the true value (solid). For the number of subgroups, we cluster particle positions using silhouette method Eqn. (10) to compute an independent value for comparison. For speed and polarization, we use particle position and velocity in Eqn. (11) to compute true values. The values of noise (η) and interaction radius (r_d) corresponding to each of the six simulations are displayed on the first row for each column. L is the size of the square domain, and ε is the root mean squared error for each feature.

Although the ground-truth data for group features is not available in this case, we use the position and velocity of each particle over time to compute the group features of structure and motion. We are not aware of a reliable method to compute the subgroup size independently; we therefore evaluate classification of images on the basis of group structure in terms of only the number of subgroups. In particular, we use the silhouette method [56] which validates a spatial partitioning scheme based on average silhouette width of the position data. For a given set of particle positions, the silhouette width w_i of each particle position $\boldsymbol{r}_i$ [56]

$$w_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \tag{10}$$

where a_i is the average distance of particle i from the members of its assigned cluster and b_i is the minimum average distance to all other clusters. Particle positions are iteratively partitioned into possible number of clusters ranging from one to N using the k-means algorithm [57], and the number of subgroups is computed as the number of clusters that maximize the average silhouette width, $\hat{w} = \sum_{i=1}^{N} w_i/N$. Figure 10 compares the number of subgroups after classification to the estimated number of subgroups based on the silhouette method at each frame for six different simulations. On MATLAB the average time to estimate the number of subgroups using the silhouette method is at least two orders of magnitude more than classification of the same image using the manifold mappings.

For classifying MHIs in terms of group motion, we use the value of group structure to obtain the corresponding group motion manifold. To evaluate the performance, we use the true position and velocity to compute the average speed $\mathcal{S}[k]$ and polarization $\mathcal{P}[k]$

$$\mathcal{S}[k] = \frac{1}{N}\sum_{i=1}^{N} s_i[k], \tag{11a}$$

$$\mathcal{P}[k] = \frac{1}{N}\Big\| \sum_{i=1}^{N} \begin{bmatrix} \cos(\theta_i[k]) \\ \sin(\theta_i[k]) \end{bmatrix} \Big\|. \tag{11b}$$

The value of $\mathcal{P}$ ranges between 0 and 1, with $\mathcal{P} = 1$ for a polarized group[1]. In each case, values of group feature obtained after classification are smoothed using a moving window average. The number of subgroups are smoothed using a moving average of window size ten and speed and polarization are smoothed using a window size T from (5), to accommodate changes in these values until they manifest in the MHI.

Figure 10 compares the values of three group features, namely number of subgroups, group speed, and group polarization as estimated after classification to those obtained by directly computing using particle trajectories. In each case we compare the result from classification to the true value using the root mean squared error, ε. For example, given the number of frames K, the error between $\hat{\mathcal{P}}$ and true value $\mathcal{P}$ of polarization is

$$\varepsilon = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(\hat{\mathcal{P}}[k] - \mathcal{P}[k])^2}. \tag{12}$$

The root mean squared error for the number of subgroups is less than two for four out of six simulations. A frame-by-frame analysis reveals that images with a few large subgroups are often classified as having more number of small subgroups, possibly due to clumping in the limited size observation region. The lowest error with respect to the number of subgroups is observed for a large noise condition where even though the particles changed configuration, they keep moving within small distinct subgroups. Since the basis for classification of motion depends on the accuracy of the number of subgroups, we find that the error propagates to speed and polarization classification. For example, the combined error in speed and polarization is relatively low despite high noise for $\eta = 0.05, r_d = \{L/80, L/40\}$. We also inspect the graphs for matching trends to verify that the classification is able to follow changes in each of these quantities. Although we do not find matching trends in speed, we observe that changes in polarization are followed for the most part. The average polarization stays high for low noise conditions ($\eta = 0.005, r_d = \{L/80, L/40\}$), and is conversely low for high noise.

4.4. Sensitivity to rotation and number of particles

Rotational variance generally causes two images with the same number of subgroups oriented differently about the center of the group to be classified as having different number of subgroups. To determine the sensitivity of classification to rotation within images, we classify test images created using the generative models after rotating them by a specific angle. The same set of three hundred test images are rotated successively by $0, 1, 2, 4, 8, 16, 32$, and 64 degrees. For each run of rotated images, the root mean square error (12) is computed for the number of subgroups. Specifically, the estimated value of the number of subgroups after classification and the true feature value used to

[1]The output of group motion classification is the range of orientations (3). We use (11b) to convert the range of orientations to group polarization.

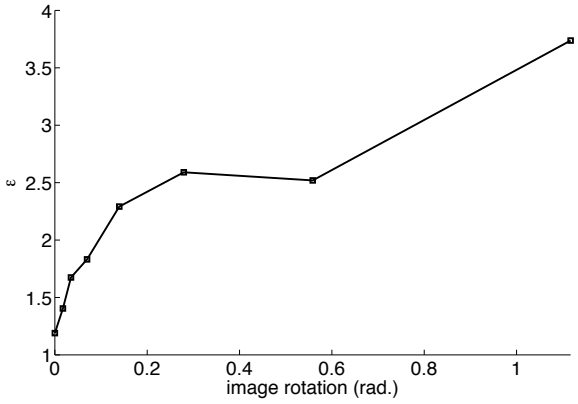

Figure 11: Root mean square error in number of subgroups for given rotation to the test image.

generate the image are compared over all frames. Figure 11 shows the root mean square error in classification of test images for number of subgroups after they are rotated.

The effect of group size on classification accuracy is also investigated. Specifically, we generate manifolds with a hundred and five hundred particles within the same square domain of size L. The maximum number of subgroups is limited to a tenth of the number of particles. (Beyond this number the finite-sized particles clump together into bigger non-distinguishable groups.) We perform two tests with each set, namely the hundred member group and five hundred member group. First, we interpolate on the manifold to verify the reconstruction error for large groups. Pixel-placement error during interpolation is 11.52 ± 4 for the hundred member group manifold and 13.45 ± 6.1 for the five-hundred member group manifold; second, we simulate a hundred and five hundred self-propelled particles at two different noise levels $\eta = \{0.05, 0.5\}$ and $r_d = L/80$ for five hundred time-steps. The root mean squared error ε is computed over all frames for the difference between the classified value of the number of subgroups and the same computed using the silhouette method (10). This error (with respect to η) is 5.20 (with respect to 0.05) and 2.39 (with respect to 0.5) for the hundred member group. Corresponding values for the five hundred member group are 5.23 (with respect to 0.05) and 2.08 (with respect to 0.5).

5. Identifying group structure on video of schooling fish

Here, we adapt the tools presented in this work to demonstrate the analysis of real images of animal groups. In a final step to illustrate the possibility of using generative models and nonlinear manifold learning to classify videos of collective behavior, we implement the methods described in this paper to analyze videos of schooling zebrafish (*Danio rerio*) [36] based on the number of subgroups. The videos recorded at 12 frames per second with a resolution of 1920×1080 pixels are available as part of supplementary material in [36]. The frames match a far-field view simulated in the training images for group structure and motion generated in Section 2. (A separate manifold is generated for different number of targets.) The video shows groups of zebrafish schooling in a circular tank. While the small groups such as five and ten fish stay together, larger groups tend to break into visually distinct subgroups as well as distribute sparsely within the tank.

Group structure manifolds with 5, 10, 20, 30, and 50 particles are generated using 500 training images with the same range parameters as in Section 2 and the value of $\kappa = 4$. The value of λ is varied as before until a two-dimensional manifold is obtained in Isomap. Specifically, $\lambda \approx 2$ for smaller groups of 5 and 10 and $\lambda \approx 1$ for group sizes of 20, 30, and 50. The number of radial basis centers, N_q, are set to thirty percent of the number of points.

To match the training images, we crop each frame to isolate the tank and rescale to 100×100 pixels. Next we extract the foreground comprising fish only by building a running background as follows. Given the image frame $I[k]$ at time k, we use the knowledge that the foreground (fish) are darker than the background to compute the background image at pixel location (u, v)

$$B_{u,v}[k] = \max(B_{u,v}[k-1], I_{u,v}[k]), \tag{13}$$

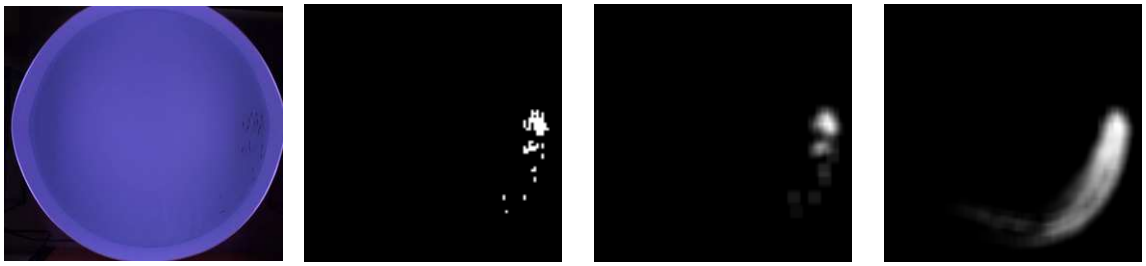

Figure 12: Real images obtained from supplementary material of [36] are processed to emphasize the group features. (a) raw image from a video of twenty zebrafish, (b) binary foreground after background subtraction, (c) blurred foreground after convolving with a Gaussian kernel to highlight group structure, and (d) motion history image using immediate prior frames to highlight group motion.

where maximizing on the intensity value at each pixel ensures that the brightest pixel is selected from all the frames upto k. A binary foreground at time k is generated at each pixel (u, v) as

$$F_{u,v}[k] = \max(I_{u,v}[k] - B_{u,v}[k], b) - b, \tag{14}$$

where b is the threshold on intensity. Following the procedure from Section 2.2, we emphasize the position by blurring the foreground, and highlight the movement by building the motion history images (Fig. 12). Note that all these operations are computationally quick to perform.

We verify the classification visually by sorting the frames according to the number of subgroups in a representative video set of $5, 10, 20, 30$, and 50 fish from [36]. The foreground images are projected onto the group structure manifold using the inverse mapping ${}^{S}f^{-1}$. Figure 13 shows five equally spaced frames from the complete sorted sequence of the videos such that the number of subgroups in a row increase from left to right. A clear progression from a cohesive group to a loosely spaced group is seen in videos with $10, 20, 30$, and 50 fish. To demonstrate that the algorithm picks up changes in the number of subgroups, a 25 second subsection of the video with twenty fish is sorted similarly in figure 14 with ten equally spaced frames.

To compare the performance of the algorithm with group properties that would be obtained from tracking individual fish, we extract fish position in each frame by locating the centroid of each blob in the foreground image. (The blobs are isolated in the foreground using the *regionprops* routine in MATLAB.) Details on the tracking algorithm can be found in [67]; note that, manual repair is not performed in this instance since occlusions automatically denote a subgroup membership and therefore do not alter the number of subgroups. Fish positions in pixels are converted to cm by calibrating the camera view. Candidate clusters ranging from 1 to the number of targets in the video are input to the silhouette method and the number of clusters that maximize the silhouette width is selected as the ground-truth.

Figure 15 compares the number of subgroups computed by applying the silhouette method on the fish position and that obtained using the method described in this paper. Table 1 shows the root mean square error ε on every tenth frame for each video. The error scaled by the numbers of targets is less than 20% indicating that the method is successful in isolating subgroups from real experimental data without the need for computationally expensive tracking [67]. However, the error with respect to real videos is in general larger than the one observed with simulated large groups in Section 4.4. We attribute such error to low variability in group structure arrangement in training data that can cause an isolated fish in large groups to be interpreted as a subgroup during classification. This can be partially addressed by increasing the uncertainty in (2) with an accompanying augmentation of the training dataset.

6. Discussion

The generative modeling and manifold learning process presented in this paper provides a computationally efficient alternative to the task of classifying images of collective animal behavior on the basis of high-level features. We classify images of particle groups on the basis of features highlighting spatial distribution and group movement. The two-step classification of images, where we first identify group structure and then group motion, is similar to the problem of separating style and content in human pose recognition [58, 14], where the content (pose) is different for

Table 1: Root mean square error ε for ten different videos from [36]

Video	No. of targets	ε
01	8	1.569
02	8	1.253
03	8	1.148
04	8	1.624
05	8	1.606
06	5	0.875
07	10	1.201
08	20	1.986
09	30	4.537
10	50	8.088

each style (human shape and form). In the case of animal groups the content (group motion) is different for a given style (group structure). Based on a selection of parameters, such as the size and resolution of the video, and the extent of a group member on the image, generative models are quickly able to create an exhaustive dataset of training images without the need for experimentation and labeling. Differently from a Bayesian approach, where the features are extracted by nonlinear estimation [37], we use the Isomap algorithm, which is a non-probabilistic technique known to effectively handle nonlinearities in the input space [19]. For example, this allows for processing the images to emphasize group features while maintaining a simple generative model. Another advantage of the classification on a frame-by-frame basis using manifold learning techniques is that error does not propagate in time (see Fig. 10, where initial large errors in the case of $\eta = 0.005, r_d = L/40$ decrease).

We find that the manifolds of group structure and motion do not form on the same axes as the input features used to produce training images (see Figures 5 and 6). This implies that the principal directions (axes) of the manifold as reconstructed by Isomap are not necessarily the same as the input features, but instead a nonlinear function of the same. In applications of manifold learning for visualization of high-dimensional data, the axes are typically inferred by correlating known properties of images with observed variations on the manifold [19], however, in our case such variations do not stand out immediately. Despite the absence of the knowledge of principal variations, we demonstrate that it is possible to extract meaningful information from the manifold provided that it satisfies few conditions which are amenable to verification. These are the presence of minimal shortcuts and a smooth embedding curve. Typically, dimensionality reduction using Isomap is sensitive to κ, the number of nearest neigbhors, and there are several methods in the literature that aim to locate the optimal choice [59]. While we do not claim to have addressed the problem of sensitivity to κ, we do provide a supervision step by successively increasing the temporal weighting parameter λ to obtain an acceptable low-dimensional embedding for a for a given value of κ.

Results from interpolation and classification using test data offer evidence of the accuracy of the manifold-to-image and manifold-to-feature mappings. Interpolation shows the capability of generating new images that vary along salient features of groups using the manifold-to-image mapping (see Figures 7a and b). Classification of test data demonstrates the ability to detect features from raw images. Classification on the basis of number of subgroups, group speed and range of orientations show good accuracy given that these images are not selected for the learning process (see Figures 8, 9). We also find that the accuracy of classification depends on the smooth variation between successive images. For example, the error in classification of MHIs of group motion which consist of blurred lines slowly moving apart is less than classifying foreground images of group structure, where individual particles change positions abruptly when the number of subgroups vary.

Detailed analysis of the sources of error in classification reveal high sensitivity to rotation in images. Upon comparison of classification error with test images and rotated versions thereof, we note that the error stays within an acceptable value of until 32 degrees after which it rises sharply. Rotational invariance, a desirable feature, can be achieved by using a rotationally invariant distance function to compare the images [60]. Alternatively, adding the group orientation about the center as a third feature to the input space may also address the problem at the cost of

increasing storage requirements. The generative model assumes equally spaced subgroups. While the model can be easily adjusted to vary the distance between subgroups as shown in Appendix A, we do not observe any change in performance by using the variable distance representation. This is because the MDS algorithm uses the distance matrix to build the manifold, whose isometric properties do not change by varying the distance between subgroups. Finally, interpolation and classification of larger particle sets reveal that the technique scales well for datasets six times and even thirty times larger than the sixteen particle set on which most of the analysis is performed.

The ability to differentiate between images along specific features of group behavior may also be enhanced by using alternate distance functions. In this paper, we use a Euclidean norm of the difference of vectorized image matrices (6). Alternate methods include diffusion distance [61, 62], normalized cross-correlation [63], and the earth mover's distance [64]. We find that histogram based distance functions, such as diffusion distance, undermine the generative model since the final image representation is an intractable function of the input feature, thereby making it difficult to control the variability in the training dataset. When we compare training images using histograms the Isomap algorithm shows inconsistency in obtaining the true embedding (data not reported). We find normalized cross-correlation and earth movers' distance to be computationally expensive operations for processing large datasets of training images. In ongoing work, we are investigating image representations and distance function that highlight features of group behavior and are robust to noise.

The ability of the manifold learning method to classify real images is demonstrated in the videos with varying numbers of schooling fish where (a) the images are successfully sorted according to the number of subgroups, and (b) a comparison between the number of clusters as computed by running silhouette method on fish position and that obtained from classification follow the same general trend. (Note that although neither of the methods used to compute the number of subgroups—silhouette method and classification—match the number of subgroups that are perceived visually, each follows the same trend of detecting a spatial increase in the degree of clustering near the middle of the video.) In our tests, we also found that due to high group density, whereby members stay close together making them indistinguishable, training images with a different number of particles could be used to classify videos, showing that this method is robust to small changes in group size. Finally, the computational advantage in using this method to analyze videos of collective motion make it an attractive tool for interactive experiments [29, 65] with animal groups, which would otherwise need sophisticated tracking algorithms and high computational power for online operation.

7. Acknowledgements

Sachit Butail and Maurizio Porfiri are supported by the National Science Foundation under grants nos. CMMI-1129820 and CMMI-0745753. Erik M. Bollt is supported by the National Science Foundation under grant no. CMMI-1129859. The authors acknowledge constructive feedback from the anonymous reviewers and discussions with Nicole Abaid.

Appendix A. Generative model for group structure

Generative models for group structure should exhaustively sample the space of features that we seek to classify in real images, namely the number of subgroups and the size of each subgroup. One way to generate subgroups is to locate them on a circle. Given an N member group, the position of i-th member $\boldsymbol{r}_i^j$, belonging to the j-th subgroup, whose center is $\boldsymbol{c}_j$, is generated as

$$\boldsymbol{r}_i^j = \boldsymbol{c}_j + \gamma_2 \begin{bmatrix} \cos(2\pi(i-1)/(N_j-1)) \\ \sin(2\pi(i-1)/(N_j-1)) \end{bmatrix} \delta(i) + \boldsymbol{\xi}_s, \text{where} \tag{A.1a}$$

$$\boldsymbol{c}_j = \frac{L}{4} \begin{bmatrix} \cos(2\pi(j-1)/(\gamma_1-1)) \\ \sin(2\pi(j-1)/(\gamma_1-1)) \end{bmatrix} \delta(j), \text{ and} \tag{A.1b}$$

$$\delta(x) = \begin{cases} 0 & \text{if } x = 1 \\ 1 & \text{otherwise.} \end{cases} \tag{A.1c}$$

The integer number of members in subgroup j is $N_j = \mathrm{int}(N/\gamma_1)$, and the level of confidence in the model is denoted by the noise $\boldsymbol{\xi}_s$ sampled from a zero-mean Gaussian distribution with standard deviation $\boldsymbol{\sigma}_s$; $\boldsymbol{\xi}_s$ adds variability to the

position of members within a subgroup. If increased, the level of noise tends to de-emphasize the spatial clustering of the subgroups, eventually leading to a structure that appears similar to having a full uniform distribution. Additionally, variability in the subgroup centers can be added in the form of noise $\boldsymbol{\xi}_c$ with standard deviation $\boldsymbol{\sigma}_c$ to (A.1b) such that $\boldsymbol{\sigma}_s < \boldsymbol{\sigma}_c$.

Appendix B. Manifold learning using Isomap

To perform classification of images on a reduced-dimensional space of the embedding manifold an image-to-manifold mapping is required so that input images are projected and located on the manifold space. However, such a mapping requires interpolation on the high-dimensional image space that is impractical to implement as it needs a large number of points [14]. In the case when the image-to-manifold mapping is invertible, we can instead generate the manifold-to-image mapping $f : \mathbb{R}^e \to \mathbb{R}^d$ using radial basis functions. The manifold is approximated by radial basis functions of the form $\phi(\|\boldsymbol{q}_i - \boldsymbol{y}\|)$ where $\boldsymbol{q}_i, i = 1, \ldots, N_q$ are points (not necessarily on the manifold) called radial basis centers. These centers (approximately twenty percent of the number of points on the manifold) can be computed using k-means algorithm. The mapping function from the e-dimensional centers to the p-th dimension in the image space, is approximated using a combination of basis functions $\phi(\|\boldsymbol{y} - \boldsymbol{q}_i\|)$ with corresponding weights w_i [14]

$$\begin{aligned} &f^p : \mathbb{R}^e \to \mathbb{R}, \\ &= \ell^p(\boldsymbol{y}) + \sum_{i=1}^{N_q} w_i^p \phi(\|\boldsymbol{y} - \boldsymbol{q}_i\|), \end{aligned} \tag{B.1}$$

where ℓ^p is a linear polynomial in $\boldsymbol{y}$. The basis function may be linear ($\phi(x) = x$) or radial ($\phi(x) = \sqrt{x^2 + a^2}$), where a is a constant [66, 14]). The combined $d \times (N_q + e + 1)$ dimensional mapping, $\mathcal{B}$, from the manifold to the input image space in matrix form is

$$f(\boldsymbol{y}) = \mathcal{B}\psi(\boldsymbol{y}), \tag{B.2}$$

where

$$\psi(\boldsymbol{y}) = \begin{bmatrix} \phi(\|\boldsymbol{y} - \boldsymbol{q}_1\|), \ldots, \phi(\|\boldsymbol{y} - \boldsymbol{q}_{N_q}\|) & 1 & \boldsymbol{y}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{B.3}$$

$\mathcal{B}$ consists of the unknown weights w_i^p and the coefficients of the polynomial ℓ^p. With enough input images ($> N_q + e + 1$), (B.2) can be set up as a system of linear equations and solved using a least-squares. The manifold-to-feature mapping is derived in the same way with $d = 2$ and by mapping the functions to corresponding values in the feature space.

References

[1] T. Vicsek, A. Zafeiris, Collective motion, Physics Reports 517 (2012) 71–140.

[2] P. Romanczuk, M. Bär, W. Ebeling, B. Lindner, L. Schimansky-Geier, Active Brownian particles, The European Physical Journal Special Topics 202 (2012) 1–162.

[3] T. A. Frewen, I. D. Couzin, A. Kolpas, J. Moehlis, R. Coifman, I. G. Kevrekidis, Coarse collective dynamics of animal groups, in: Coping with Complexity: Model Reduction and Data Analysis, 2011, pp. 299–309.

[4] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, V. Zdravkovic, Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study, Proceedings of the National Academy of Sciences of the United States of America 105 (2008) 1232–1237.

[5] W. Choi, K. Shahid, S. Savarese, What are they doing? : Collective activity classification using spatio-temporal relationship among people, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, 2009, pp. 1282–1289.

[6] T. A. Patterson, M. Basson, M. V. Bravington, J. S. Gunn, Classifying movement behaviour in relation to environmental conditions using hidden Markov models., The Journal of Animal Ecology 78 (2009) 1113–23.

[7] R. Li, R. Chellappa, Group motion segmentation using a Spatio-Temporal Driving Force Model, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2038–2045.

[8] J. Delcourt, C. Becco, N. Vandewalle, P. Poncin, A video multitracking system for quantification of individual behavior in a large fish shoal: Advantages and limits, Behavior Research Methods 41 (2009) 228.

[9] J. K. Parrish, W. M. Hammer, Animal groups in three dimensions, Cambridge University Press, 1997.

[10] Y. Katz, K. Tunstrø m, Inferring the structure and dynamics of interactions in schooling fish, Proceedings of the National Academy of Sciences of the United States of America 108 (2011) 18720–18725.

[11] J. E. Herbert-Read, A. Perna, R. P. Mann, T. M. Schaerf, D. J. T. Sumpter, A. J. W. Ward, Inferring the rules of interaction of shoaling fish., Proceedings of the National Academy of Sciences of the United States of America 108 (2011) 18726–31.

[12] L. Cayton, Algorithms for manifold learning, Technical Report, University of California, San Diego, 2005.

[13] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, The Journal of Machine Learning Research 7 (2006) 2399–2434.

[14] A. Elgammal, C.-S. Lee, Nonlinear manifold learning for dynamic shape and dynamic appearance, Computer Vision and Image Understanding 106 (2007) 31–46.

[15] J. Blackburn, E. Ribeiro, Human motion recognition using Isomap and dynamic time warping, in: Proceedings of Conference on Human motion: understanding, modeling, capture and animation, pp. 285–298.

[16] M.-H. Yang, Face recognition using extended isomap, in: Proceedings of the International Conference on Image Processing, volume 2, IEEE, 2002, pp. 117–120.

[17] C. BenAbdelkader, Robust head pose estimation using supervised manifold learning, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 518–531.

[18] R. Pless, Image spaces and video trajectories: using Isomap to explore video sequences, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, 2003, pp. 1433–1440.

[19] J. B. Tenenbaum, V. de Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction., Science 290 (2000) 2319–23.

[20] M. Kirby, Geometric data analysis: an empirical approach to dimensionality reduction and the study of patterns, Wiley, 2001.

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

[22] L. K. Saul, S. T. Roweis, Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds, Journal of Machine Learning Research 4 (2003) 119–155.

[23] J. Wang, Z. Zhang, H. Zha, Adaptive manifold learning, Advances in Neural Information Processing Systems (2004).

[24] T. F. Cox, M. A. Cox, Multidimensional scaling on a sphere, Communications in Statistics - Theory and Methods 20 (1991) 2943–2953.

[25] N. Abaid, E. Bollt, M. Porfiri, Topological analysis of complexity in multiagent systems, Physical Review E 85 (2012) 041907.

[26] P. DeLellis, M. Porfiri, E. Bollt, Topological analysis of group fragmentation in multi-agent systems (to appear), Physical Review E (2013).

[27] M. H. C. Law, A. K. Jain, Incremental nonlinear dimensionality reduction by manifold learning., IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 377–391.

[28] R. Souvenir, R. Pless, Image distance functions for manifold learning, Image and Vision Computing 25 (2007) 365–373.

[29] V. Kopman, J. Laut, G. Polverino, M. Porfiri, Closed-loop control of zebrafish response using a bioinspired robotic-fish in a preference test, Journal of the Royal Society Interface 10 (2013) 20120540.

[30] V. D. Silva, J. B. Tenenbaum, Global Versus Local Methods in Nonlinear Dimensionality Reduction, Advances in Neural Information Processing Systems 15 (2003).

[31] R. Mann, R. Freeman, M. Osborne, R. Garnett, C. Armstrong, J. Meade, D. Biro, T. Guilford, S. Roberts, Objectively identifying landmark use and predicting flight trajectories of the homing pigeon using Gaussian processes., Journal of the Royal Society, Interface 8 (2011) 210–9.

[32] T. F. Cox, M. Cox, Multidimensional Scaling, CRC Press, 2000.

[33] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, O. Shochet, Novel Type of Phase Transition in a System of Self-Driven Particles, Physical Review Letters 75 (1995) 1226–1229.

[34] I. Couzin, J. Krause, Self-organization and collective behavior in vertebrates, Advances in the Study of Behavior 281 (2003) 17–29.

[35] S. H. Lee, H. K. Pak, T. S. Chon, Dynamics of prey-flock escaping behavior in response to predator's attack., Journal of Theoretical Biology 240 (2006) 250–259.

[36] N. Miller, R. Gerlai, From Schooling to Shoaling: Patterns of Collective Motion in Zebrafish (Danio rerio), PLoS ONE 7 (2012) e48865.

[37] C. Bishop, Pattern recognition and machine learning, Springer, 2006.

[38] I. D. Couzin, J. Krause, N. R. Franks, S. A. Levin, Effective Leadership and Decision-making in Animal Groups on the Move, Nature 433 (2005) 513–516.

[39] L. Conradt, C. List, Group decisions in humans and animals: a survey, Philosophical Transactions of the Royal Society B: Biological Sciences 364 (2009) 719.

[40] S. G. Reebs, Can a minority of informed leaders determine the foraging movements of a fish shoal?, Animal behaviour 59 (2000) 403–409.

[41] R. Vabo, L. Nottestad, An individual based model of fish school reactions: predicting antipredator behaviour as observed in nature, Fisheries Oceanography 6 (1997) 155–171.

[42] A. Czirók, T. Vicsek, Collective behavior of interacting self-propelled particles, Physica A: Statistical Mechanics and its Applications 281 (2000) 17–29.

[43] H. Levine, W. Rappel, I. Cohen, Self-organization in systems of self-propelled particles, Physical Review E 63 (2000) 017101.

[44] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, N. R. Franks, Collective memory and spatial sorting in animal groups, Journal of Theoretical Biology 218 (2002) 1–11.

[45] J. Belmonte, G. Thomas, L. Brunnet, R. de Almeida, H. Chaté, Self-Propelled Particle Model for Cell-Sorting Phenomena, Physical Review Letters 100 (2008) 248702.

[46] G. Ramos-Fernández, D. Boyer, V. P. Gómez, A complex social structure with fissionfusion properties can emerge from a simple foraging model, Behavioral Ecology and Sociobiology 60 (2006) 536–549.

[47] R. Jeanson, S. Blanco, R. Fournier, J. Deneubourg, V. Fourcassié, G. Theraulaz, A model of animal movements in a bounded space, Journal of Theoretical Biology 225 (2003) 443–451.

[48] J. Krause, G. Ruxton, Living in groups, Oxford University Press, 2002.

[49] N. Leonard, E. Fiorelli, Virtual leaders, artificial potentials and coordinated control of groups, in: Proceedings of the IEEE Conference on Decision and Control, volume 3, pp. 2968–2973.

[50] R. Pulliam, T. Caraco, Living in groups: is there an optimal group size?, in: In Behavioral Ecology: an evolutionary approach (1984), 1984, pp. 122–147.

[51] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.

[52] A. Berg, J. Malik, Geometric blur for template matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 607–614.

[53] A. Bobick, J. Davis, The recognition of human movement using temporal templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 257–267.

[54] C. Bregler, S. M. Omohundro, Nonlinear image interpolation using manifold learning, in: Advances in Neural Information Processing Systems, pp. 973–980.

[55] A. Papoulis, Probability, random variables and stochastic processes, McGraw Hill, 1991.

[56] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65.

[57] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on Mathematics, Statistics and Probability, pp. 281–297.

[58] J. B. Tenenbaum, W. T. Freeman, Separating Style and Content with Bilinear Models, Neural Computation 12 (2000) 1247–1283.

[59] O. Samko, A. D. Marshall, P. L. Rosin, Selection of the optimal parameter value for the Isomap algorithm, Pattern Recognition Letters 27 (2006) 968–979.

[60] F. Zhao, Q. Huang, W. Gao, Image Matching by Normalized Cross-Correlation, in: Proceedings of the IEEE International Conference on Acoustics Speed and Signal Processing Proceedings, volume 2, IEEE, 2006, pp. 729–732.

[61] K. Okada, Diffusion Distance for Histogram Comparison, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, IEEE, 2006, pp. 246–253.

[62] A. Kolpas, J. Moehlis, T. A. Frewen, I. G. Kevrekidis, Coarse analysis of collective motion with different communication mechanisms., Mathematical Biosciences 214 (2008) 49–57.

[63] J. Lewis, Fast normalized cross-correlation, in: Vision interface, pp. 120–123.

[64] Y. Rubner, C. Tomasi, L. Guibas, A metric for distributions with applications to image databases, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), Narosa Publishing House, 1998, pp. 59–66.

[65] J. Krause, A. F. T. Winfield, J. Deneubourg, Interactive robots in experimental biology., Trends in Ecology and Evolution 26 (2011) 369–375.

[66] D. Beymer, T. Poggio, Image Representations for Visual Learning, Science 272 (1996) 1905–1909.

[67] S. Butail, T. Bartolini, M. Porfiri, Collective response of zebrafish to a mobile robotic fish (to appear), in: Proceedings of the ASME Dynamic Systems and Control Conference.

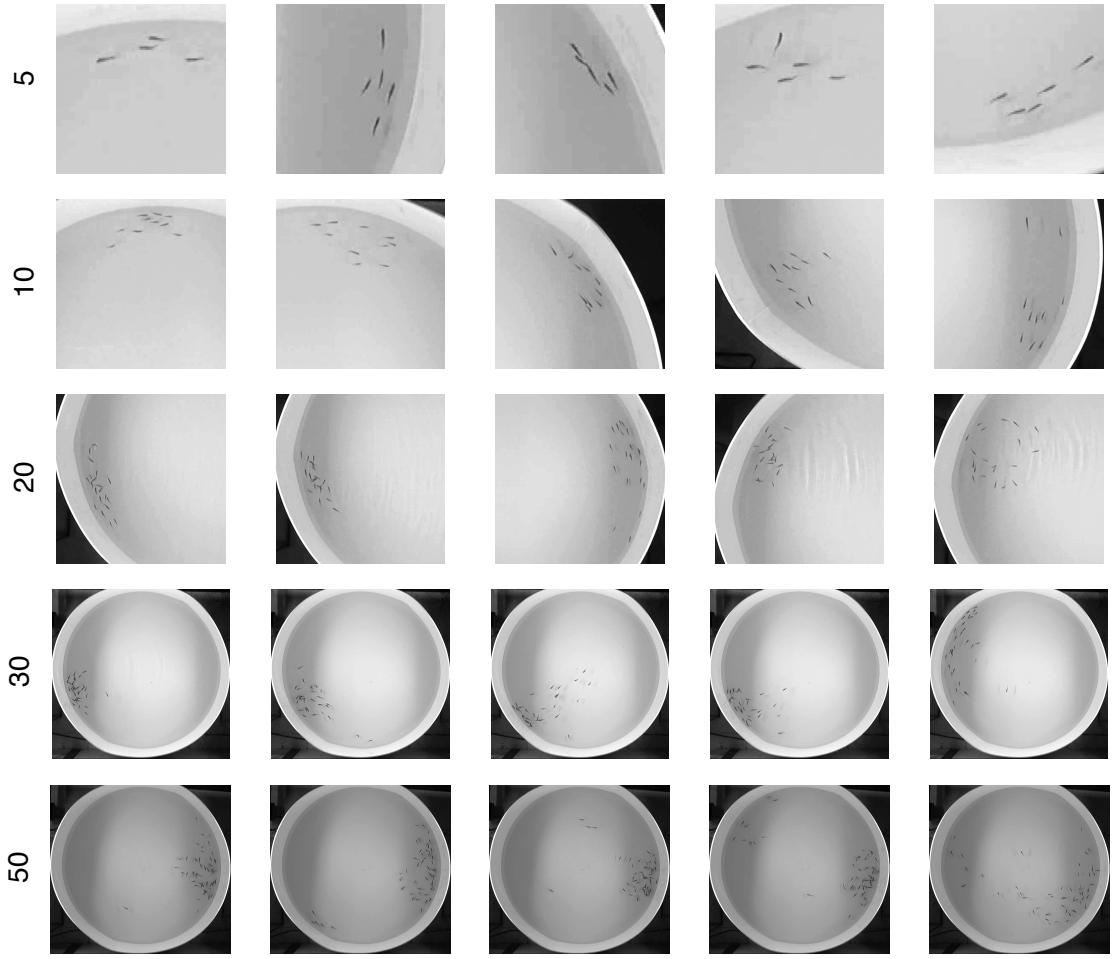

Figure 13: Frames from multiple videos sorted according to increasing number of subgroups as classified using the group structure manifold. Each row corresponds to a single video labeled with the number of fish in the same. Top three rows have the frames zoomed in by a fixed factor for each row for easier visualization.

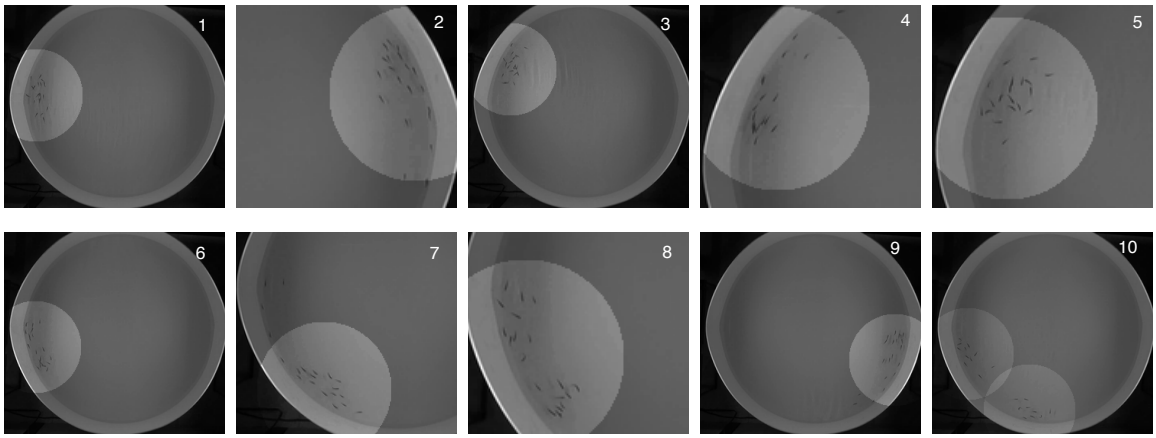

Figure 14: Foreground frames from an online video of zebrafish [36] sorted on the basis of increasing number of subgroups after projecting and classifying using the group structure manifold. The fish are highlighted by increasing contrast in a circular region around the group. Selected frames are zoomed in for easier visualization. In this 25 second subsection of the video, the group moves with a few fish trailing behind for the first 10 seconds; they then split gradually into two distinct groups until 16 seconds after which they again merge back into a single group.

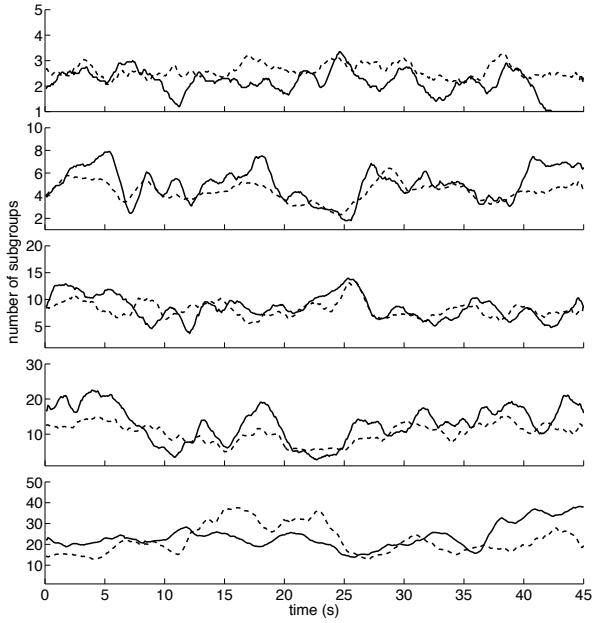

Figure 15: Comparing the number of subgroups obtained by using the silhouette method (solid) on the two-dimensional fish positions in pixels to the one obtained using the classification method described in this paper (dashed). The videos are available from literature [36]. The maximum number of subgroups in each axis is equal to the number of fish in the video. The time taken by the combined tracking and silhouette method was on average two orders of magnitude more than the classification method.