# Project B: Underwater Aggregation and Dispersion of AUVs using Event-based Detection

**Submitted By:**
Mohammed Tarnini
*Department of Mechanical and Nuclear Engineering*
100049735@ku.ac.ae

**Submitted To:**
Prof. Jorge Dias
*Department of Electrical Engineering*
jorge.dias@ku.ac.ae

***Abstract*—This study presents a simple method for detecting and estimating the state of Autonomous Underwater Vehicles (AUVs) using event-based frames. Due to the challenges of using dynamic vision sensors (DVS) in underwater environments, RGB videos were processed to emulate events using two toolboxes: v2e and SpiKeCoding. The process involves converting video frames into events, followed by preprocessing steps such as median filtering and thresholding to improve detection accuracy. The resulting events were used to estimate the relative positions of AUVs, enabling aggregation and dispersion behaviors through a Potential Field method based solely on visual data. The results demonstrate the effectiveness of this approach in creating robust, self-driven swarm behaviors for AUVs in dynamic underwater environments, with implications for decentralized control in swarm robotics.**

***Index Terms*—AUV, aggregation, dispersion, detection**

## I. Introduction

The interest in underwater robotics has significantly developed in the last decades due to the fact that 95% of the ocean has not been explored and 99% of the ocean floor is unmapped [1]. Exploring those uncharted regions opens a route for discovering new energy sources, food supplies, and medicine. Underwater robots play a very important role in many surveillance and monitoring activities such as ensuring coastal safety and collecting data on underwater ecosystems [2].

Advances in electronics have made robots lighter and more compact, essential features for underwater applications. However, it is quite challenging to conduct these tasks with the use of ROVs. Their dependency on tethering restricts movements and increases the risk of cable entanglement, hence limiting access to confined areas. Therefore, ROVs are usually deployed as single units, not as swarms, in underwater missions. Further, the use of visual data underwater is restricted due to problems such as loss of detection performance due to light refraction and visibility of short range, especially in salty water [3]. Using one robot for these tasks would take longer, which again is not feasible because most underwater systems have limited power. A swarm of Autonomous Underwater Vehicles (AUVs) intended for collective and decentralized, leaderless operation solves these issues.

Various experimental systems of AUV swarm have been developed, but the research has been relatively limited, such as [4], [5]. Underwater swarms offer enormous advantages including scalability and adaptability to dynamic environments. Each AUV relies on local information to respond and adjust in real-time, ensuring functionality even if some units fail, as the collective compensates for any losses. One example of an experimental underwater swarm implementation is the BlueBot project [4], which features a fish-inspired robotic swarm designed to mimic the collective behaviors observed in fish schools. It successfully demonstrates behaviors such as dispersion, aggregation, and milling. Collaboration among the robots was achieved through simple shared rules and visual feedback, with experiments conducted in a controlled, dark environment. Drawing inspiration from honeybee strategies, [5] proposed a decentralized algorithm for adaptive task allocation and flexible partitioning within AUV swarms. This algorithm dynamically adapts to the changes in swarm size and task demands and empowers effective swarm behaviors at different conditions without the necessity of global information.

Most of the other examples of AUV swarm applications come from simulations; [6] introduced BEELCLUST, a bio-inspired algorithm for exploration tasks based on the heat-seeking behavior of bee swarms; [5] explored AUV swarms for search-and-explore missions targeting objects in movement, relying on simulated data from visual and acoustic data; and boundary control methods were developed for navigating swarms of AUVs within defined regions of interest using potential field-based methods as in [7]. Further work by the same group extended these methods to include collision avoidance, navigation within dynamic target regions [8], and obstacle avoidance [9]. Comparable methods were presented in [10], [11], where the boundary of a spatial region approach worked effectively for pipe-following and inspection tasks. In [12], coordination over multiple groups using simple shape boundaries was shown to achieve complex formations.

The structure of the study is as follows: Section II provides an overview of the methodologies employed. Section III presents the AUV used in the experiments. Section IV presents the findings and interprets their implications. Finally, Section V outlines the conclusions drawn from the study and proposes directions for future work.

## II. METHODOLOGY

### A. Video Processing

In this study events videos are used to obtain the ranges and heading towards any detected AUV in frame. Using a DVS underwater is difficult as there are dangers of having water leaking inside the DVS and damaging the components. Hence, normal RGB videos were taken from a camera onbaord of an AUV, and this RGB video was processed to emulate events. Two toolboxes were utilized, and they are v2e [13] and SpiKeCoding [14]. Table I presents the parameters used when converting the RGB frames into events. As for SpikeCoding toolbox, the temporal contrast threshold used was 20. Both of these toolboxes were successfully used to create many events over the whole dataset acquired from RGB videos.

TABLE I
PARAMETERS OF THE V2E TOOLBOX TO GIVE CLEAN EVENTS.

| Parameter | Value |
|---|---|
| dvs_exposure | 0.033 |
| input_frame_rate | 30 |
| input_slowmotion_factor | 1 |
| pos_thres | 0.3 |
| neg_thres | 0.3 |
| sigma_thres | 0.07 |
| cutoff_hz | 10 |
| leak_rate_hz | 0.03 |
| shot_noise_rate_hz | 0.5 |

Figure 1 (a) shows an event frame obtained from v2e toolbox, and Figure 1 (b) is the event framed obtained from SpikeCoding toolbox. It is shown that even though the parameters of v2e were chosen to give clean events without noise, it is clear that SpikeCoding gave cleaner events.

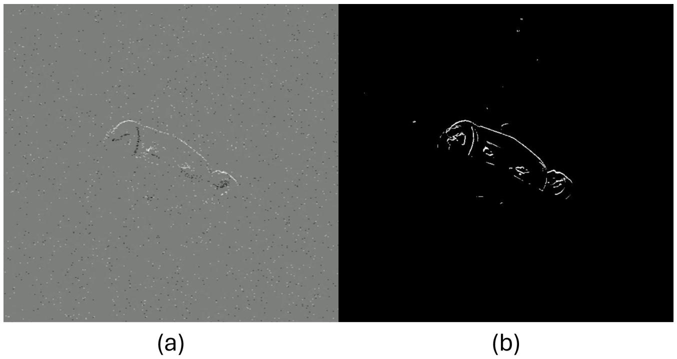

Fig. 1. (a) Event frame obtained from v2e toolbox. (b) Event frame obtained from SpikeCoding toolbox.

After obtaining events, we need to perform some preprocessing on the events to get more accurate detections. A 2D median filter *medfilt2* is used on the events obtained from v2e in order to remove the salt-and-pepper noise. The median filter works as a smoothing operator by replacing the value of each pixel by the median value of neighboring pixels; hence, this operation does not blur the image much. This step is crucial since noisy data usually reduces object detection accuracy significantly.

The events frames are transformed into grayscale by the *rgb2gray* function. This will make things easier for analysis since color information may not be required to detect the objects. Working with grayscale images reduces computational complexity because a grayscale image has only intensity values that are important for foreground detection.

A thresholding operation is done to separate the objects from the background. In this step, pixels with grayscale values above a certain threshold are classified as part of the foreground and set to white (binary value 1), while those with values below the threshold are classified as part of the background and set to black (binary value 0). This results in a binary image where the foreground objects are highlighted. The thresholding equation is:

$$\text{binaryFrame}(i,j) = \begin{cases} 1, & \text{if}\, \text{p}(i,j) > \text{threshold} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Figure 2 shows the events obtained from v2e and SpikeCoding after being processed with median filter and binary threshold. It is clear that the AUV is more visible in the SpikeCoding events when compared with v2e. This will dramatically affect the accuracy of detection as patches of white pixels are not very close to each other in v2e events. On the other hand, SpikeCoding events have a better cluster of white pixels which will make it better for detection.

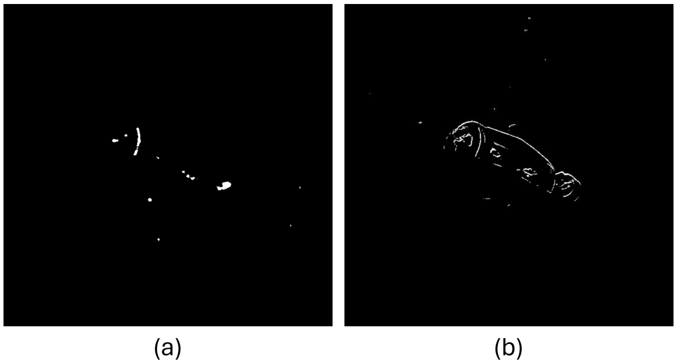

Fig. 2. (a) Processed events from v2e toolbox. (b) Processed events from SpikeCoding toolbox.

### B. Detection and State Estimation

Detection in this process begins with identifying connected components in the binary image, which represents the foreground objects after thresholding. The first step involves finding connected groups of pixels using the *bwconncomp* function. This function scans the binary image to identify all groups of neighboring white pixels that are connected either horizontally, vertically, or diagonally. Each group is treated as a distinct connected component, and the function returns a structure containing the pixel indices of these components.

Once the connected components are identified, various properties of these regions are calculated using the *regionprops* function. This function computes several characteristics for each connected component, such as the centroid, area, and bounding box. The centroid represents the center of mass of the component, calculated as the average of the pixel coordinates within the component. The area is the number of pixels that make up the component, giving a measure of its size. The bounding box is the smallest rectangle that can enclose the component, defined by its top-left corner coordinates and the rectangle's width and height. To refine the detection process, components are filtered based on their area. A predefined threshold is applied, and only those components with an area greater than the threshold are retained. This filtering step helps remove small, irrelevant objects or noise, focusing only on significant components that are more likely to be the objects of interest.

After filtering, the bounding box for each remaining component is extracted. These bounding boxes define the regions of interest in the image where the detected objects are located. The bounding boxes are stored along with the area of each component, forming a list of rectangles that represent the detected objects. This step ensures that only meaningful, sufficiently large objects are considered for further processing, such as visualization or tracking.

Using the pinhole camera model, it is possible to obtain a fairly accurate estimation of the distance to the agent based on its height. The control approach used always focuses on the closest detected agent, which corresponds to the one with the largest detected height when multiple agents are present. Let the coordinate system be centered at the center of the image. Define the center coordinates of the bounding box as $(x_c, y_c)$ and the box height as $h_b$ in pixels. The distance $d_A$ from the detected AUV to the viewer is approximated as:

$$d_{\mathrm{A}} = \frac{h_{\mathrm{A}}}{\beta} \tag{2}$$

Where where $h_A$ is the height of the AUV, and $\beta$ is the angular height of the bounding box, which is computed as:

$$\beta = \frac{h_{\mathrm{b}}}{h_{\mathrm{f}}} \cdot FOV_{\mathrm{v}} \tag{3}$$

where $h_f$ is the height of the image frame in pixels, and $FOV_v$ is the vertical field of view of the camera measured in radians.

Another important parameter is the angular correction in the horizontal field of view, given by:

$$\Delta\theta = \frac{x_{\mathrm{c}}}{w_{\mathrm{f}}} \cdot FOV_{\mathrm{h}} \tag{4}$$

where $w_f$ is the width of the image frame in pixels, and $FOV_h$ is horizontal field of view of the camera, measured in radians. These parameters are visualized in 3. The values $d_A$ and $\Delta\theta$ will be used in the control loop to perform aggregation and dispersion behaviors.

### C. Aggregation and Dispersion Method

We use the Potential Field method, which will allow us to aggregate or disperse only on visual information. The ability of robots to aggregate or disperse in a self-driven manner at any location in the environment or disperse over the available area, respectively, is possible only due to this method. The underlying idea behind it is that the relative position between the robots determines the attractive or repulsive force applied between the two. A commonly used potential function for aggregating robots is the Lennard-Jones function, below:

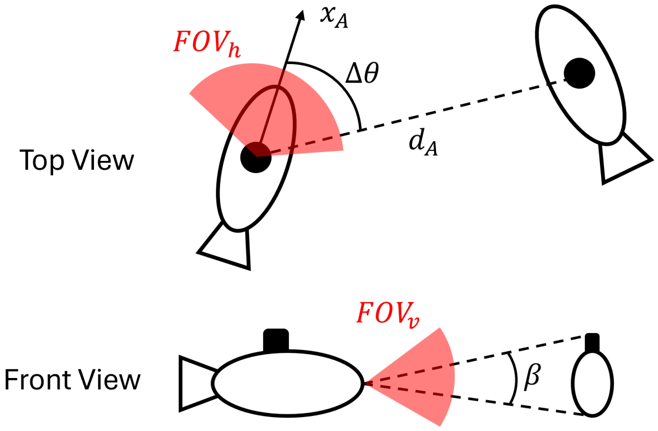

Fig. 3. Representation of the state estimation parameters using top view and front view of two AUVs

$$V = 4\varepsilon \left[ \left(\frac{\sigma}{r}\right)^{2n} - \left(\frac{\sigma}{r}\right)^{n} \right] \tag{5}$$

Here, $\varepsilon$ is the depth of the potential function, $\sigma$ is the distance at which the potential value is zero, $n$ defines the exponent that controls the steepness of both the repulsive and attractive forces, and $r$ represents the distance between two robots. The potential function describes the interaction between robots, where the force acting on each robot is derived by differentiating the potential with respect to $r$. This derivative gives a force equation that will govern the motion of the robots, either attracting for aggregation or repelling for dispersion. The force equation is given by:

$$F = \frac{dV}{dr} = 4\varepsilon \left[ -2n \left( \frac{\sigma^{2n}}{r^{2n+1}} \right) + n \left( \frac{\sigma^{n}}{r^{2n+1}} \right) \right] \tag{6}$$

This force defines the behavior of the robots in the system. When the objective is aggregation, an attractive force will be exerted on the robots, drawing them to a center point where they come together. Conversely, if dispersion is the objective, a repulsive force will be exerted on the robots, driving them apart to uniformly cover the environment. By relying solely on visual data, robots can execute such behaviors in the absence of external communications or complex navigation systems. Therefore, this method turns out to be very useful for decentralized systems.

## III. Platform Overview

The AUV that was used for obtaining images and detections is seen in Figure 4. Each AUV is equipped with a 9-DoF

IMU to calculate its absolute heading, accompanied by a pressure sensor for depth measurements. The Agents have 3 controllable DoFs, actuated by 3 brushless motors. This makes the system underactuated: surge and yaw are actuated by two motors placed on the aft, while a third vertical motor, mounted at the bottom, controls heave. It prescribes the Agents' movement input, surge force, absolute heading, and depth from the model and simulations. Such inputs are then converted by low-level control, discussed in [15] to motor commands. Each Agent is also equipped with an acoustic modem for communication with the other Agents and the Floater-Sinker. The modem could serve for giving command to the AUVs to perform the aggregation or dispersion behaviors.

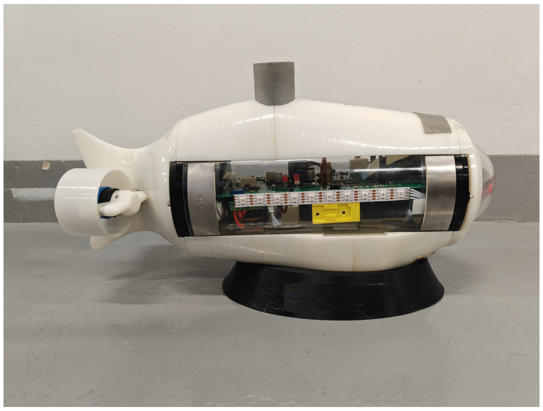

Fig. 4. H-SURF AUV, a single Agent. Note the 2 rear propellers on the back providing surge and heading correction.

The command given for an AUV to move consists of surge force, absolute heading, and absolute depth. In this case, the AUVs are assumed to only move on the horizontal plane, so the command needed for the behaviors requires only surge force and absolute heading. The force command and heading command are obtained from the vector summation of all forces that arise from detections of AUVs.

## IV. Results and Discussion

### A. Detections

Figure 5 illustrates the detection results using the same algorithm, which employs binary thresholding as previously described. In Figure 5 (a), the frame generated by v2e contains highly scattered patches, leading to significant occlusion of the AUV. Consequently, the bounding box around the detected object is notably small. 100 random frames from the dataset were used to calculate the average Intersection Over Union (IOU) and Mean Average Precision at 40% threshold (mAP@0.4) and both of them were almost 0. On the other hand, Figure 5 (b) demonstrates detection using a frame obtained from SpikeCoding, where the object is more distinctly visible, enabling a more accurate and appropriately sized bounding box to be drawn around it. Also here, 100 random frames were used to calculate the average IOU and mAP@0.4. The average IOU obtained was 0.316, and mAP@0.4 was 0.41.

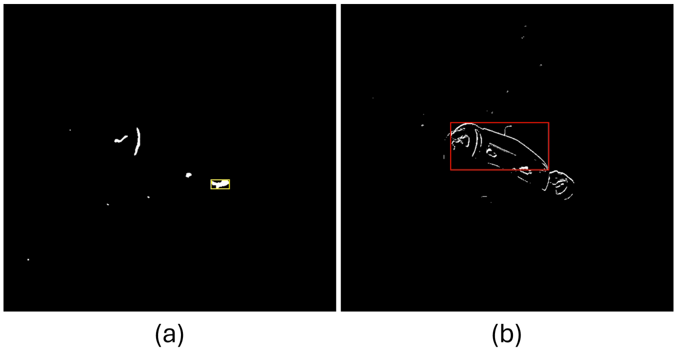

Fig. 5. Detections on frames from v2e events shown in (a) and SpikeCoding shown in (b).

The low values of average IOU and mAP@0.4 are expected, because the viewer is stationary, and the AUV moving was quite slow, so the white pixels were not always visible in frames. In real conditions where the viewer is moving and both AUVs have high speeds, the accuracy of detections will obviously increase. Another reason for these low values are the presence of obstacles in the frames where they sometimes appear as white pixels and decrease the accuracy of detection. Despite all this, the accuracy is still reasonable and the detection can still be used to get suitable values of heading and ranges for aggregation and dispersion behaviors.

### B. Aggregation

Aggregation simulation was conducted for 500 seconds inside a square boundary, as shown in Figure 6. The number of AUVs used in this setup was 16, positioned at random inside the square boundary. The inter-AUV distance desired was set at 1 meter. The AUVs were aggregated at some random position within the boundary due to attractive forces calculated using the Lennard-Jones potential function. During the simulation, the AUVs formed a compact cluster, with an average spacing close to the target distance of 1 meter. Their orientations remained random, consistent with the absence of specific constraints on heading in the aggregation model. A fencing behavior was also incorporated to actively prevent the AUVs from crossing the boundary during the aggregation process. This combination of attractive forces and boundary enforcement successfully facilitated the desired aggregation behavior within the simulated environment.

During the simulation, the density of AUVs was calculated at each time step. Density is defined number of AUVs contained within a convex volume defined by the outermost AUVs, which can be used to evaluate the aggregation motion. Density started with a value close to 0.2 at the beginning of the simulation due to the random initialization of AUV positions. Then, the density kept increasing to reach a value of about 1.2 in 100 seconds, where the ideal value is 1 due to the desired distance between AUVs being 1. This shows how quick the AUVs aggregate and the high accuracy of aggregation.

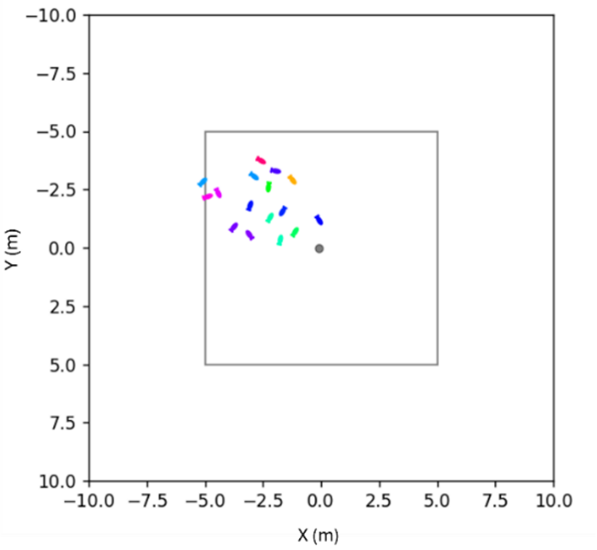

Fig. 6. Aggregation with fencing inside a square boundary

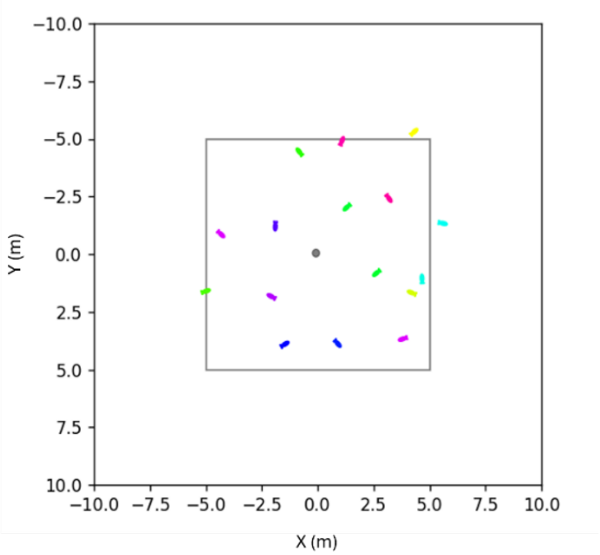

Fig. 7. Dispersion with fencing inside a square boundary

### C. *Dispersion*

Dispersion simulation was conducted for 500 seconds inside a square boundary, as shown in Figure 7. The number of AUVs used in this setup was 16, positioned at random inside the square boundary. It is clear that the AUVs are spread almost uniformly inside the square boundary. For dispersion, the Lennard-Jones potential function was adjusted to only give a repulsive force whenever an AUV is detected. Similar to aggregation the orientation of AUVs is random even when all AUVs are spread uniformly, consistent with the absence of specific constraints on heading in the dispersion model. A fencing behavior was also incorporated to actively prevent the AUVs from crossing the boundary during the aggregation process. This combination of repulsive forces and boundary enforcement successfully facilitated the desired dispersion behavior within the simulated environment.

Detections of AUVs underwater were successful which facilitated the implementation of aggregation and dispersion behaviors on an AUV swarm. Following that, aggregation and dispersion simulations using Lennard-Jones Potential Function were successful and showed quick convergence towards the desired behavior. **Supplementary Videos are attached to this report** to show the successful tests done in regards of detection, aggregation, and dispersion.

## V. Conclusion

The challenges in using dynamic vision sensors underwater made the authors decide to emulate events from RGB videos captured by onboard cameras. They used two toolboxes, V2E and SpikeCoding, to convert RGB frames into events, with fine-tuning of parameters for best performance. SpikeCoding had better performance, giving much cleaner event frames that raised the accuracy of detection significantly. The median filtering and binary thresholding preprocessing steps were performed on the event data to enhance the quality of detection. These steps have reduced noise and highlighted objects, thus enabling AUV identification. A pinhole camera model was used for distance estimations and angular corrections; these allowed for accurate control in aggregation and dispersion behaviors. By utilizing the potential field method, the AUVs realized decentralized behaviors relying only on visual information and further demonstrated the robustness and adaptability of the system toward achieving collective behaviors.

The extension to a more complicated underwater scene is a next step within the current framework. Scalability shall be studied experimentally, including multiple environmental conditions-light and turbidity conditions, among other factors. Besides, the incorporation of advanced object detection and state estimation through machine learning could further improve the accuracy and efficiency of the proposed approach. These features have the potential to make underwater swarm robotics revolutionary, enabling robust and autonomous operations in dynamic and challenging environments.

## References

[1] Y. Yang, Y. Xiao, and T. Li, “A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability,” *IEEE Communications Surveys and Tutorials*, vol. 23, pp. 815–841, April 2021.

[2] B. Das, B. Subudhi, and B. B. Pati, “Cooperative formation control of autonomous underwater vehicles: An overview,” *International Journal of Automation and Computing*, vol. 13, pp. 199–225, June 2016.

[3] Q. Jia, H. Xu, X. Feng, H. Gu, and L. Gao, “Research on cooperative area search of multiple underwater robots based on the prediction of initial target information,” *Ocean Engineering*, vol. 172, pp. 660–670, January 2019.

[4] F. Berlinger, M. Gauci, and R. Nagpal, “Implicit coordination for 3d underwater collective behaviors in a fish-inspired robot swarm,” *Science Robotics*, vol. 6, January 2021.

[5] P. Zahadat and T. Schmickl, “Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition,” *Adaptive Behavior*, vol. 24, pp. 87–101, April 2016.

[6] M. Bodi, C. Möslinger, R. Thenius, and T. Schmickl, “Beeclust used for exploration tasks in autonomous underwater vehicles,” *IFAC-PapersOnLine*, vol. 48, pp. 819–824, 2015.

[7] C. Cheah and Y. Sun, “Adaptive region control for autonomous underwater vehicles,” in *Proceedings of the IEEE*, 2004, pp. 288–295.

[8] C. C. Cheah, S. P. Hou, and J. Slotine, “Region following formation control for multi-robot systems,” in *Proceedings of the IEEE*, May 2008, pp. 3796–3801.

[9] S. P. Hou and C. C. Cheah, “Can a simple control scheme work for a formation control of multiple autonomous underwater vehicles?” *IEEE Transactions on Control Systems Technology*, vol. 19, pp. 1090–1101, September 2011.

[10] Z. H. Ismail and M. W. Dunnigan, “A region boundary-based control scheme for an autonomous underwater vehicle,” *Ocean Engineering*, vol. 38, pp. 2270–2280, December 2011.

[11] Z. H. Ismail, N. Sarman, and M. W. Dunnigan, “Dynamic region boundary-based control scheme for multiple autonomous underwater vehicles,” in *Proceedings of the IEEE*, May 2012, pp. 1–6.

[12] R. Haghighi and C. Cheah, “Multi-group coordination control for robot swarms,” *Automatica*, vol. 48, pp. 2526–2534, October 2012.

[13] Y. Hu, S.-C. Liu, and T. Delbruck, “v2e: From video frames to realistic dvs events,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1312–1321.

[14] J. Dupeyroux, S. Stroobants, and G. C. De Croon, “A toolbox for neuromorphic perception in robotics,” in *2022 8th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*. IEEE, 2022, pp. 1–7.

[15] S. Iacoponi, G. J. V. Vuuren, G. Santaera, N. Mankovskii, I. Zhilin, F. Renda, C. Stefanini, and G. De Masi, “H-surf: Heterogeneous swarm of underwater robotic fish,” in *OCEANS 2022, Hampton Roads*. IEEE, 10 2022, pp. 1–5.