2012 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 23–26, 2012, SANTANDER, SPAIN

HIDDEN MARKOV MODELS FOR DETECTING ANOMALOUS FISH TRAJECTORIES IN UNDERWATER FOOTAGE

C. Spampinato, S. Palazzo

Department of Electrical, Electronics and Computer Engineering
University of Catania, Viale Andrea Doria, 6 - 95127 Catania, Italy
Email: {cspampin, palazzosim}@dieei.unict.it

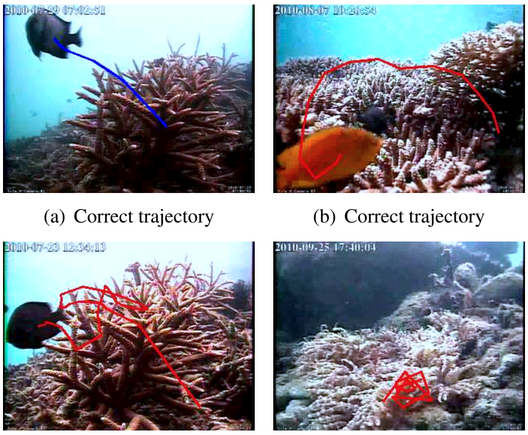
ABSTRACT

In this paper we propose an automatic system for the identification of anomalous fish trajectories extracted by processing underwater footage. Our approach exploits Hidden Markov Models (HMMs) to represent and compare trajectories. Multi-Dimensional Scaling (MDS) is applied to project the trajectories onto a low-dimensional vector space, while preserving the similarity between the original data.

Usual or normal events are then defined as set of trajectories clustered together, on which HMMs are trained and used to check whether a new trajectory matches one of the usual events, or can be labeled as anomalous. This approach was tested on 3700 trajectories, obtained by processing a set of underwater videos with state-of-art object detection and tracking algorithms, by assessing its capability to distinguish between correct trajectories and erroneous ones due, for instance, to object occlusions, tracker mis-associations and background movements.

1. INTRODUCTION

The recent advances in camera technology and the decrease of costs of devices have led to a widespread interest towards computer vision applications for automatic video processing in fields such as video surveillance [1] and real-life animal studies [2, 3]. However, although a lot of research effort has been put in the development of algorithms for object detection and tracking, the problem of identifying and understanding high-level behavioural events from low-level motion information is still unsolved. Several approaches have been proposed by the scientific community [4, 5], but, given the variable nature of the task and its dependance to each specific application context, it is extremely difficult to define a globally valid method. Typically, these techniques follow a bottom-up approach, consisting in mapping low-level motion features, such as speed, direction, size and appearance variations, or an object's whole trajectory, to high-level scene semantics, by either recursively identifying simple events and combining them into more complex ones, or by using automatic clustering techniques to identify sets of "common" events, which can then be mapped to specific behaviours or used to identify "anomalous" events, i.e. events that do not fall into any of the patterns identified as usual. However, most of the existing approaches are targeted, as expected, at humans, while the analysis of fish trajectories in an unconstrained environment shows many more problems. First of all, the quality of the trajectories themselves is lower than in other contexts, because the technical difficulties related to capturing, transmitting and storing underwater videos often limit the resolution and frame rate of the images. Also, scene conditions can be difficult to handle, because of lighting variations, clearness of the water, presence of moving background objects (e.g. algae). Moreover, fish have more degrees of freedom than humans, their bodies are not rigid, so appearance models have to be able to handle fast shape transformations, and finally they have a typical erratic motion which makes them difficult to follow. Examples of fish trajectories are shown in Figure 1.



(a) Correct trajectory

(b) Correct trajectory

(c) Complex but correct trajectory

(d) Wrong trajectory, due to background movement

Fig. 1. Examples of fish trajectories. 1(a) and 1(b): simple trajectories; 1(c): complex but still correct trajectory; 1(d): a wrong trajectory, due to background plant movements.

In this paper, we propose an approach to identify uncommon trajectories of fish when processing real-life underwater videos. Our approach is based on the method proposed in [6] and uses Hidden Markov Models (HMMs) to encode and compare trajectories and, afterwards, projects the trajectories onto a low-dimensional space which preserves the similarity of trajectories through Multi-Dimensional Scaling (MDS). The resulting vectors are then fed to a K-means clustering algorithm in order to identify common motion patterns (i.e. common "events") which are used to identify incorrect trajectories, generated as a result of errors of object detection and tracking algorithms. The results showed that our method performs well in the detection of incorrect trajectories, while keeping the false positive rate (i.e. the percentage of regular trajectories identified as anomalous) to acceptable levels.

The remainder of the paper is as follows: Section 2 briefly introduces the state of the art on event detection through object trajectory analysis; Section 3 describes the proposed method for trajectory representation and clustering; whereas Sections 4 and 5 show, respectively, the experimental results when the method was applied to fish trajectories and the considerations and ideas for future improvements and applications.

2. RELATED WORKS

Given the importance of the possible applications that behaviour-understanding and anomaly-detection algorithms can have, as mentioned in the previous paragraph – one for all, in video surveillance –, many approaches have been proposed in the literature for trajectory analysis. Basically, these approaches can be divided into two main categories:

- Automatic recognition of common/uncommon events, e.g. by clustering trajectories and learning motion patterns.
- Recursive composition of events, from low-level motion information to high-level scenarios.

The main advantage of the first category of approaches is that they apply well-known data clustering methods to automatically extract common patterns, thus minimizing the effort required to define behaviours and events. Of course, these approaches are very sensitive to how trajectories are represented and how the learning algorithm is tuned (for example, the number of clusters can be a fundamental choice for the success or failure of the approach). Relevant examples of these approaches are:

- In [4], the authors apply a grammar rule induction framework to learn event rules. A clustering approach based on [7] is used to identify simple motion patterns. Hidden Markov Models (HMMs) are trained to model each cluster, and are used as detectors of primitive events. A grammar induction algorithm, where grammars are evaluated according to the Minimum Description Length (MDL) principle [8], is then applied to build the set of event rules.
- Porikli *et al.* [5] propose a method for the detection of unusual events based on spectral clustering. Histograms and HMMs based on objects' speed, color, size, aspect ratio, etc are used as features for trajectory description. For each feature, an affinity matrix (where the (i, j)-th element shows how similar the i-th and j-th objects are, according to that feature) is built and then decomposed using a certain number of the largest eigenvalues. After further transformations, a correlation matrix is computed, and clustering consists in grouping the elements which result highly correlated.
- In [9], a clustering method for trajectories is presented, which can be applied both to improve tracking performance (by predicting the position of an object at time $t + 1$ according to the best-matching cluster at time t) and to detect anomalous trajectories (by evaluating how frequently each cluster is matched, and considering clusters with few elements as "anomalous"). In this approach, clusters actually represent relatively short segments of trajectories, so each trajectory can be made up by segments belonging to different clusters, organized in a tree structure.

Other approaches are based on a semantic reconstruction of the scene, by recognizing just simple events at first (such as "being still" or "moving in a certain direction"), and then combining them, spatially and/or temporally, into more complex events (for example, "approaching", "following", or a combination of simultaneous sub-events). Unlike the automatic approaches, it is necessary to explicitly define rules for the description of scenarios, i.e. these algorithms are manually tuned by the users according to the scenarios they deal with and this, of course, limits their applicability. For example, Medioni *et al.*, in [10], use trajectory data and a-priori information on the scene to define three abstraction levels in the event recognition process: image features (size, speed, position, distance from reference objects), mobile object properties (entering a certain area, approaching reference objects or other actors, etc), scenarios (combinations of mobile object properties or, recursively, other scenarios). Similarly, Cupillard *et al.* in [11] model the different scenarios with "basic properties" (trajectory, speed, etc), states (a situation which involves a set of actors at a certain time, or which holds for a certain period) and events (variations of states).

3. TRAJECTORY ANALYSIS THROUGH HMM CLUSTERING

The approach we used in our analysis of fish trajectories takes inspiration from the work proposed in [6]. This method belongs to the "clustering" category, which is one of the reasons that makes it suitable to study fish trajectories, since it is difficult to define events in terms of a sequence of simple moves, because of the characteristics of fish motion. In detail, we have extended the original approach by making it able to learn motion patterns of any target (instead of only humans) through the trajectory's spatial and temporal features only. Moreover, we have also corrected/changed the metrics used to perform the Multi-Dimensional Scaling since the original ones were not clearly explained and seemed to be wrong.

The basic idea is to use Hidden Markov Models (HMMs) to represent trajectories in a uniform way, without having to deal with different path sizes while keeping the underlying trajectory's dynamics. A metric for HMMs is introduced in order to build a similarity matrix between all objects in the learning set which is used by a Multi-Dimensional Scaling (MDS) algorithm to project trajectories onto a lower-dimensional space, where it is more feasible to perform trajectory clustering to identify classes corresponding to common patterns. In order to decide whether a new trajectory is anomalous, for each cluster a corresponding HMM is built and used to check whether the input trajectory matches it; trajectories that do not match significantly any of all the identified clusters are therefore detected as anomalous.

Figure 2 shows the flowchart of the system. Each component of the whole system is described in detail in the next subsections.

3.1. Modeling trajectories as HMMs

One of the main problems in the analysis of object trajectories is finding an appropriate way of representing them. The typical point-sequence representation, although it contains all the information describing the movement of an object, is often difficult to work with, since a comparison of different-length trajectories requires a normalization of the number of points, with the risk of over- or under-sampling; moreover, for the same reason, it is difficult to represent a generic motion pattern as a sequence of points. Histograms of position, speed, orientation (e.g. [5]), etc. may also be employed to describe trajectories, but they lose all temporal information, which is an essential part of the pattern recognition process. On the contrary, HMMs are often used in the description of trajectories, since they can intrinsically encode spatio-temporal sequences of data and also provide intuitive algorithms to generate sample trajectories and to check whether an input trajectory matches the pattern learned by the HMM.

A Hidden Markov Model (HMM) is a stochastic model describing a Markovian process where the states are not directly observable, differently from a regular Markov chain. The estimation of the current state is then performed by analyzing the system's output variables, which depend on the current state: assuming discrete output variables, each state has a probability distribution over the values these variables can assume, hence by analyzing the output sequences it is possible to obtain the information necessary for the estimation of the state sequence. HMMs can be trained from output sequences, making them especially feasible for temporal pattern recognition [12]. The parameters of an n-state HMM with m discrete output variables are:

- Prior distribution π: probability for the initialization of the HMM's first state.
- State transition probabilities A: an $n \times n$ matrix whose $a_{i,j}$ element is the probability of going from state i to state j.
- Emission distributions B: an $n \times m$ matrix whose $b_{i,j}$ element is the probability that, in state i, the output token will be j.

The set of the three model matrices is typically referred to as λ. Of course, the structure and dimensions of these matrices can vary if there are multiple output variables or if the distribution is continuous, as is the case of this work. A description of continuous-output HMMs using mixtures of Gaussians is presented in [12]. The proposed method models each trajectory (of a initial training set) with an HMM, whose output variables are position coordinates, speed and direction of the object, modeled by mixtures of Gaussians. Differently from the original approach, we do not force the states of the model to match real world locations, instead we let the HMM learn its own internal configuration by applying the Baum-Welch algorithm and feeding a trajectory or a set of trajectories as input. Moreover, we do not apply the state transition probability defined by the authors because it does not hold for 3D unconstrained motion such as fish movement. All states have the same initial probability.

After HMM training, trajectories are projected into a more compact space which is more suitable for the subsequent clustering phase.

3.2. MDS projection and clustering

In order to represent the trajectories in a format more appropriate to K-means clustering than HMMs, Multi-Dimensional Scaling (MDS) [13] is applied, which projects HMMs to a relatively short vector space, while maintaining the distances in the original HMM space. Since MDS exploits a distance matrix among input data to reduce the original space, we have introduced a probabilistic metric to compare HMMs describing trajectories. If λ_1 and λ_2 are the parameters which model trajectories O_1 and O_2, we adopt Juang and Rabiner's approach [14] in defining the (asymmetric) distance $D\left(\lambda_1, \lambda_2\right)$ as:

$$D\left(\lambda_1, \lambda_2\right) = \left[log\, L\left(O_1|\lambda_1\right) - log\, L\left(O_1|\lambda_2\right)\right] \qquad (1)$$

where $L\left(O_x|\lambda_y\right)$ is the probability that trajectory O_x is modeled by λ_y. Since equation (1) is not symmetric, an averaged

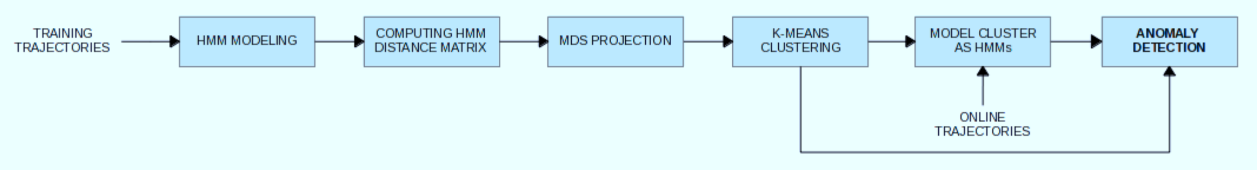


Fig. 2. A flowchart of the proposed system.

distance is computed as:

$$D_{ave}\left(\lambda_1, \lambda_2\right) = \frac{1}{2}\left(D\left(\lambda_1, \lambda_2\right) + D\left(\lambda_2, \lambda_1\right)\right) \qquad (2)$$

In order to avoid errors with short data sequences, the HMM parameters λ_k for a single trajectory are interpolated (linear interpolation between the π, A, and B matrices) with λ_{all}, representing the HMM parameters obtained by training a model with all trajectories in the training set:

$$\begin{aligned} \pi_k &\leftarrow \beta\pi_k + \left(1-\beta\right)\pi_{all} \\ A_k &\leftarrow \beta A_k + \left(1-\beta\right)A_{all} \\ B_k &\leftarrow \beta B_k + \left(1-\beta\right)B_{all} \end{aligned} \qquad (3)$$

The MDS algorithm takes as input a distance matrix D, where each element $d_{i,j}$ is equal to $D_{ave}\left(\lambda_i, \lambda_j\right)$. Starting from D, a B matrix (which projects the original points into points whose barycentric coordinates are the origin) is computed, whose eigenvalues and eigenvectors are then employed to compute the projected trajectory vectors (details are described in [13]).

These vectors are then clustered using unsupervised K-means algorithm [15]. We perform two clustering cycles: one to filter out outliers from the training dataset and the second one to build the clusters of common paths which are then used to detect anomalous trajectories. In detail the executed steps are:

- First clustering of the HMMs modeling the input trajectories (training data set) into k clusters.
- Train k HMMs with the trajectories in each cluster.
- For each trajectory, detect it as anomalous if the maximum likelihood between it and the k HMMs is lower than a threshold.
- Re-clustering of the input trajectories training data set into k clusters, leaving out the trajectories identified as anomalous by the previous step. This step makes the final clusters more accurate and with less variance around their centroids.

The resulting k HMMs are then used to evaluate the likelihood that a test trajectory belongs to one of the common path clusters, and if the maximum likelihood is smaller than a threshold, the trajectory is labeled as anomalous.

4. APPLICATION TO FISH TRAJECTORIES

We have applied the approach described in Section 3 to underwater domain, as part of the Fish4Knowledge[1] project, which aims at providing marine biologists with a reliable system for fish detection, tracking and behavioural analysis.

In order to extract fish trajectories from these videos, we have employed an object detection algorithm [16], able to handle all the changes that may affect the analysed scene, and a covariance-based tracker [17]. Although the performance of the overall system is good, it is unavoidable that errors sometimes occur, for example because of occlusions between fish and/or objects in the scene and of errors in the object detection algorithm, which identifies moving background objects, such as plants, as fish. For these reasons, the proposed system has been applied also to identify erroneous trajectories (i.e. not corresponding to feasible fish motion patterns) computed by the tracker. Of course, such trajectories will be labeled as “anomalous”, since the HMMs representing each cluster will have been trained with correct trajectories only.

4.1. Experimental results

We performed several tests in order to investigate the best configuration of the HMM/K-means system to identify incorrect trajectories. The training set T_{train} was made up of 300 correct fish trajectories with lengths between 5 and 30 points, manually selected from the tracking results on 10 underwater videos (320×240 resolution at 5 fps) from the F4K repository. The test set used to evaluate the system consisted of 3700 trajectories, with uniformly distributed lengths between 3 and 50, equally divided into correct and erroneous trajectories.
Since the output results depend on the HMMs' parameters we performed a parametric analysis of the performance in order to find the best HMM configuration for the target we are dealing with. The first HMM parameter we analyzed was the number of iterations for the Baum-Welch algorithm used to train the models. We found out that, for HMMs built from a reasonable number of trajectories (for example, those representing whole clusters), 45 iterations was a good compromise between computation time and the maximization of the probability that trajectories in the training set matched

[1] *http://www.fish4knowledge.eu*

HMM configuration			Results	
S	*M*	*T*	*DR*	*FAR*
5	4	-100	55.6%	36.7%
5	4	-120	21.0%	3.1%
5	16	-100	57.3%	32.7%
5	16	-120	24.3%	7.1%
15	4	-100	55.0%	28.61%
15	4	-120	23.9%	2.0%
25	4	-100	54.2%	29.6%
25	4	-120	26.1%	5.1%
10	4	-100	18.9%	1.0%
10	4	-120	22.4%	0.0%
15	16	-100	59.6%	29.5%
15	16	-110	39.2%	13.3%

Table 1. Performance of the system with different HMM configurations.

that HMM. However, for HMM representing single trajectories, this value resulted to be too high, because states transition probabilities would "flatten", practically making states equiprobable. Moreover, whichever smaller number of iterations we applied, the input data sequence was too short to make the resulting HMM effectively learn the trajectory pattern. The solution we found to this problem was to train the single-trajectory HMM with 15 iterations, and interpolating it with the HMM parameters obtained by learning all trajectories in the training set.

After setting the number of training iterations, the next parameter we analyzed was the number of clusters. We ran a few simulations, with varying HMM parameters (the number of states and output mixtures), and applied the method described in [15] to estimate the optimal number of clusters. According to most of these simulations, 3 clusters were enough to represent the variability of the training set (and no simulations gave a result smaller than 2 or larger than 4), so we set the number of clusters for the following tests to 3.

Our first test session consisted in varying HMM parameters to gather some preliminary information on the most promising configuration. Table 1 shows the best performance of the system we achieved by varying HMM parameters: in the left group of columns, *S* is the number of HMM states, *M* is the number of output mixtures and *T* is the minimum log-likelihood threshold used to decide whether a given trajectory matches at least one cluster; on the right columns, for each configuration we show the corresponding detection rate (*DR*, the percentage of correctly identified anomalous trajectories) and the false alarm rate (*FAR*, the percentage of correct trajectories identified as anomalous).

The results we obtained with these first tests showed the difficulty in choosing a set of HMM parameters which provided satisfactory percentages of both true positives and false negatives. The best HMM configuration seemed to be the ones with 15 states and 16 mixtures of Gaussians, and the most discriminating variable was clearly the probability threshold.

Length range	*T*	*DR*	*FAR*
≤10	-80	88.8%	4.3%
11-15	-100	80.2%	14.2%
16-20	-160	85.4%	29.2%
21-25	-190	96.2%	16.7%
≥30	-230	82.4%	16.6%

Table 2. Performance of the system when the threshold is varied according to the length of the training and test trajectories.

In order to understand if the HMM parameters were somehow related to the trajectory length, we carried out a second run of experiments which consisted in building several test sets, each containing the test set trajectories having lengths included in a certain range (for example, $T_{test,\leq 10}$ contained the test trajectories with up to 10 points, $T_{test,11-15}$ contained trajectories with 11 to 15 points, and so on), and finding for each of them the best probability threshold. The HMM's numbers of states and number of output mixtures were 15 and 16, since these values yielded the best results in the previous test. Table 2 shows the best threshold for each range, and the corresponding detection rate and false alarm rate for the each of the new test sets.

It is clear how the trajectory length actually influences the capability of HMMs to match them, and how a length-depending threshold allows to obtain a much higher accuracy than using a fixed threshold.

5. CONCLUDING REMARKS

The problem of inferring semantic events from a low-level analysis of motion information in videos is still far from solved, because of the obvious difficulties in finding a common model to different contexts and because of the variety of possible targets and event types. Many approaches, based on a recursive recognition of simple and complex events or on clustering trajectories into common patterns, have been proposed in scientific literature.

In this paper we investigated the possibility of applying a method for the identification of anomalous events, specifically thought for humans, to fish trajectories, computed by processing underwater videos with state-of-art motion detection and tracking algorithms. This approach, based on a representation of trajectories through Hidden Markov Models and on the computation of clusters representing common motion patterns, is able to identify anomalous trajectories, defined as those which show significant differences from all the above-mentioned clusters.

The evaluation of this approach was performed by assessing its capability to distinguish valid fish trajectories from incorrect ones, computed by the object detection and tracking but affected by errors (e.g. misassociations, occlusions, background object movements). The results show the the method is effectively able to distinguish between correct and incorrect trajectories, thus providing a useful tool for automatic error identification and object detection/tracking improvements.

Of course, the generality of the method and of the definition of what can be an "anomalous" event allows for an extension of the purposes for which it can be applied. For the next step in our work, we will investigate the possibility to map the clusters of usual events to specific behaviours, such as feeding, preying, mating, etc. However, this task requires prior knowledge on how fish show these behaviours, which is not as evident as with humans. Therefore, a preliminary behaviour ground-truth generation stage will have to be performed, since it has not been tackled yet in the literature for such targets.

6. ACKNOWLEDGEMENTS

This research was funded by the European Commission (FP7 grant 257024) and the Taiwan National Science Council (grant NSC100-2933-I-492-001) and undertaken in the Fish4Knowledge project (*www.fish4knowledge.eu*).

7. REFERENCES

[1] A. Faro, D. Giordano, and C. Spampinato, "Integrating location tracking, traffic monitoring and semantics in a layered its architecture," *IET Intelligent Transport Systems*, vol. 5, no. 3, pp. 197–206, 2011.

[2] C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. B. Fisher, "Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos." in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP 2008)*, 2008, pp. 514–519.

[3] C. Spampinato, D. Giordano, R. D. Salvo, Y. J. Chen-Burger, R. B. Fisher, and G. Nadarajan, "Automatic fish classification for underwater species behavior understanding," in *First ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*, 2010, pp. 45–50.

[4] Z. Zhang, K. Huang, T. Tan, and L. Wang, "Trajectory Series Analysis based Event Rule Induction for Visual Surveillance," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Jun. 2007.

[5] F. Porikli and T. Haga, "Event Detection by Eigenvector Decomposition Using Object and Frame Features," *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

[6] N. Suzuki, K. Hirasawa, K. Tanaka, Y. Kobayashi, Y. Sato, and Y. Fujino, "Learning motion patterns and anomaly detection by Human trajectory analysis," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*. Ieee, 2007, pp. 498–503.

[7] M. Meila and J. Shi, "A random walks view of spectral segmentation," in *AI and Statistics AISTATS*, vol. 2001, AISTATS. Citeseer, 2001, pp. 8–11.

[8] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, ser. Series in computer science ; vol. 15. World Scientific, 1989, vol. 15.

[9] C. Piciarelli, G. L. Foresti, and L. Snidaro, "Trajectory clustering and its applications for video surveillance," *Advanced Video and Signal Based Surveillance, IEEE Conference on*, 2005.

[10] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 873–889, 2001.

[11] F. Cupillard, A. Avanzi, F. Bremond, and M. Thonnat, "Video understanding for metro surveillance," in *IEEE International Conference on Networking Sensing and Control 2004*, vol. 1. Ieee, 2004, pp. 186–191.

[12] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[13] G. Young and A. S. Householder, "Discussion of a set of points in terms of their mutual distances," *Psychometrika*, vol. 3, no. 1, pp. 19–22, 1938.

[14] B. H. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden Markov models," *AT&T Technical Journal*, vol. 64, no. 2, pp. 391–408, 1985.

[15] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics Theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[16] O. Barnich and M. Van Droogenbroeck, "ViBe: a universal background subtraction algorithm for video sequences." *IEEE Transactions on Image processing*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[17] C. Spampinato, S. Palazzo, D. Giordano, F. P. Lin, and Y. T. Lin, "Covariance-based fish tracking in real-life underwater environment," in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP 2012)*, 2012.