Machine Vision and Applications
DOI 10.1007/s00138-013-0509-x

SPECIAL ISSUE PAPER

A rule-based event detection system for real-life underwater domain

Concetto Spampinato · Emmanuelle Beauxis-Aussalet · Simone Palazzo · Cigdem Beyan · Jacco van Ossenbruggen · Jiyin He · Bas Boom · Xuan Huang

Received: 26 August 2012 / Accepted: 2 April 2013

Abstract Understanding and analyzing fish behaviour is a fundamental task for biologists that study marine ecosystems because the changes in animal behaviour reflect environmental conditions such as pollution and climate change. To support investigators in addressing these complex questions, underwater cameras have been recently used. They can continuously monitor marine life while having almost no influence on the environment under observation, which is not the case with observations made by divers for instance. However, the huge quantity of recorded data make the manual video analysis practically impossible. Thus machine vision approaches are needed to distill the information to be investigated. In this paper, we propose an automatic event detection system able to identify solitary and pairing behaviours of the most common fish species of the Taiwanese coral reef. More specifically, the proposed system employs robust low-level processing modules for fish detection, tracking and recognition that extract the raw data used in the event detection process. Then each fish trajectory is modeled and classified using hidden Markov models. The events of interest are detected by integrating end-user rules, specified through an ad hoc user interface, and the analysis of fish trajectories. The system was tested on 499 events of interest, divided into solitary and pairing events for each fish species. It achieved an average accuracy of 0.105, expressed in terms of normalized detection cost. The obtained results are promising, especially given the difficulties occurring in underwater environments. And moreover, it allows marine biologists to speed up the behaviour analysis process, and to reliably carry on their investigations.

Keywords Event detection · Fish detection · Fish recognition · Trajectory classification · Behaviour understanding

This research was funded by European Commission FP7 grant 257024, in the Fish4Knowledge project (www.fish4knowledge.eu).

C. Spampinato (✉) · S. Palazzo
Department of Electrical, Electronics and Computer Engineering, University of Catania, Catania, Italy
e-mail: cspampin@dieei.unict.it

S. Palazzo
e-mail: simone.palazzo@dieei.unict.it

E. Beauxis-Aussalet · J. van Ossenbruggen · J. He
Centrum voor Wiskunde en Informatica (CWI), Amsterdam, Netherlands
e-mail: Emmanuelle.Beauxis-Aussalet@cwi.nl

J. van Ossenbruggen
e-mail: Jacco.van.Ossenbruggen@cwi.nl

J. He
e-mail: jiyin.he@cwi.nl

C. Beyan · B. Boom · X. Huang
School of Informatics, University of Edinburgh, Edinburgh, UK
e-mail: C.Beyan@sms.ed.ac.uk

B. Boom
e-mail: bboom@inf.ed.ac.uk

X. Huang
e-mail: Xuan.Huang@ed.ac.uk

1 Introduction

The continuous progress in digital cameras and information storage capacities, with consequent cost reduction, led to an exponential proliferation of video-surveillance applications, specifically developed for investigating events and behaviours in human-centered applications [1–3]. This explosion also generated new possibilities for investigating diverse domains, such as fauna monitoring, through the use of embedded cameras which have low impact and

Published online: 04 May 2013

low interference with the natural environment [4–6]. In fact, nonintrusive visual observation plays a crucial role for researchers in domains such as ecology that address complex questions about natural environments and about the behaviours and interactions of their living organisms.

An interesting example is the EcoGrid project[1] which collected many terabytes of videos with the aim of monitoring forest animals and fish living in Taiwan. And other examples can be found with cameras filming animals such as bird nests, wolves, or foxes.

However, it is unrealistic to assume people can fully investigate all the generated videos, because it requires a lot of time and concentration while being error prone. Therefore, machine vision techniques are highly envisioned for automatically mining such data. The Fish4Knowledge project[2] contributes to this direction. It develops video processing tools to support marine biologists that study fish populations of the coral reefs of Taiwan's shores.

Marine biologists are mainly interested in specific or unusual behaviours, such as migration, preying, schooling, or mating. Their analysis of the changes in behaviour patterns, and of the behaviour characteristics of different species, allows them to detect and study the environmental conditions, such as pollution and climate change [7,8].

To address their needs, we propose here an automatic event detection system. It is able to recognize specific fish behavioural events by detecting fish activities and fish interactions in the videos collected for the Fish4Knowledge project. More specifically, the events of interest are identified by integrating user-defined rules with the analysis of fish behaviours modeled using hidden Markov models. Although in the literature there exist many approaches for event detection (as reviewed in the next section), they are mainly tailored for different usages in domains ranging from sport to airport monitoring for security purposes. To our knowledge, our proposed system is one of the first approaches addressing behavioural event detection for the underwater domain. A previous attempt was proposed by Spampinato et al. [9] but it aimed at detecting environmental events (e.g. typhoons and storms), and at investigating how fish behaviour changes when such events occur.

In addition to the event detection system, we propose an intelligent user interface (UI) that supports and speeds up the ground-truth annotation process. The Fish4Knowledge video repository contains over 500,000 videoclips that are 10 min long. It is practically impossible to manually analyze this amount of videos for selecting the video excerpts needed for the training of the machine vision classifiers and for the performance evaluation. The UI supports the definition of rules that allow the retrieval of potentially relevant video excerpts. User can select and label these excerpts, and use them as the ground-truth for event detection. The user-defined rules target co-occurrence of fish, since biologists are mostly interested in detecting activities involving solitary fish and pairs of fish.

The main contributions of our paper to the discussion on event detection in video-surveillance are the following:

- First, we evaluate the applicability of object detection, recognition and trajectory modeling approaches on noisy environments affected by low video quality (e.g., due to limit in network bandwidth), background sudden changes (e.g., sudden light changes due to the gleaming of the sun on the water), massive presence of background objects (e.g. plants, corals).
- Second, we propose a motion-based event detection system which exploits trajectories extracted in a complex real-life use case. Indeed, our videos are more cluttered and denser than the ones usually tackled in human video-surveillance systems. Moreover, fish movement is more complex than human's, since fish are featured by non-rigid and erratic motion. Finally, we encounter partial or total overlapping of fish due to the unconstrained motion in three dimensions.
- Finally, we propose a UI for video labeling that supports users in the complex task of collecting event ground-truth. It uses the output of the low level processing modules (i.e., the fish detection, tracking an recognition) to facilitate the exploration of videos. The user-defined rules allow to retrieve fish co-occurrences of interest, in order to distill the number of videos to be inspected and labelled by users.

In the remainder of the paper, Sect. 2 reviews the mainstream approaches for event detection in human-centered applications. Section 3 describes the flowchart of our event detection system, describing in detail the components employed for fish detection, tracking and recognition, trajectory modeling and classification. Section 4 presents the user interface for video labeling, the species-specific events it targets, and their meaning for marine biologists. Sections 5 and 6, respectively, discuss the performance of our system, and the concluding remarks and ideas for future developments.

2 Related works

Many different event detection approaches have been proposed in literature. They can vary regarding the following points:

- *Targets* humans have been the typical object targeted by event detection algorithms. Depending on the kind of

[1] http://ecogrid.nchc.org.tw/.

[2] www.fish4knowledge.eu.

event that is targeted, the difficulty of the task can be relatively easy (e.g. changing speed, changing direction, chasing [10]) or hard (e.g., a specific action, such as using an object or waving at people [11]). Another kind of target is urban vehicles, e.g., for traffic monitoring, or for anomalous and illegal behaviour detection [12]. This kind of applications are usually easier than those involving humans, since vehicles move on constrained paths and their appearance does not change a lot. Another case study are animals. In this case, besides implying motion dynamics that differ from human ones, complications arise from the technical difficulties in filming wildlife and from the complexity of representing animal-related events [13].

- *Context* the two main application fields that were investigated by the research community are video-surveillance (e.g. [10,14]) and sports (e.g., [15,16]). In video-surveillance, cameras are typically stationary, and the strongest requirements concern the risk of false alarms: no one would want a bank video-surveillance system to alert the police every time something passes by the field of view of the camera. In sport, the scene's background is changing, both because of camera movements and because of scene cuts. Thus the estimation of the absolute position of players is difficult. Further, the scene switchings themselves may provide information on what happens in the match [15].
- *Event recognition approach* when the application domain and the kinds of event to detect are completely known, some approaches explicitly define an event in terms of combination of low-level motion properties or simple actions [10], and apply heuristic rules [17] or finite state machines [18]. Other approaches [19,20] apply machine learning techniques (e.g., hidden Markov models [19] or dynamic Bayesian networks [21]) to learn typical event patterns, given the features used to describe the targets' actions (e.g., people trajectories). The former approach allows to explicitly describe the types of event of interest, thus making the detector more accurate. However, this approach can detect *only* the events it was instructed for, and it requires that the targeted events are defined, and definable, as a sequence of easy-to-detect sub-events. The latter approach, on the other hand, automatically learns how to recognize events of any kind (e.g., by clustering trajectories, or learning motion patterns). It has the advantage to discover unseen data patterns, thus providing results for scientists to investigate. But since the events are inferred by the algorithm, the main disadvantage is that end-users must be particularly careful when selecting the algorithms and their parameters. This condition how the approach could fit a specific context.

In this section, we will describe a few examples of state-of-the-art event detection algorithms, varying among all of the above-mentioned targets, contexts and algorithm types.

2.1 Targets

The typical target of event detection algorithms are people, since most practical applications involve the analysis of human actions, ranging from security to entertainment. However, depending on the kind of events into consideration, the way people are modeled can vary.

In some applications, the simple identification of a moving person is sufficient for the event detection purpose. In this case it consists of an analysis of trajectories to find those which match, or do not match, a learnt pattern of anomalous behaviours. This strategy also applies to vehicle behaviour analysis, and is typical of a video-surveillance application. For example, in [22], 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 by considering clusters with few elements as "anomalous"). In this approach, clusters actually represent relatively short segments of trajectories and are organized in a tree structure. Trajectories can be decomposed in segments belonging to different clusters. In Porikli et al. [20], histograms and hidden Markov models (HMMs) based on objects' features (such as speed, color, size, aspect ratio) are used for trajectory description. This allows to integrate temporal information in the description of motion.

In other applications, the events to detect consist of specific actions performed by a human. In this case, a trajectory-based representation of the human involved is not sufficient. The single limbs may have to be detected and tracked and, even harder, matched against a known moving pattern. In [11], the authors present a work on the detection of short events, such as picking objects or waving hands, in crowded environments. It typically involves difficulties due to interferences between humans. A spatio-temporal segmentation is performed on the video and is compared with the templates describing the events of interest. The template-matching engine is able to detect parts of event (sliced both spatially and temporally), for a more robust recognition. The main limit is that the event model is built from a single example, with no fusion mechanism implemented for defining a template from multiple samples.

Event detection in a *crowd* involved different approaches to analyze human behaviour. In crowd analysis, from a computer vision point of view, the techniques used for individual targets are not appropriate, since the concepts of "motion detection" and "tracking" acquire a different meaning.

Nevertheless, there are suitable techniques [23] that allow to tackle the event detection problem. For example in [24], the authors compute the crowd optical flow and use unsupervised feature extraction to encode normal crowd behaviour. Spectral clustering is applied to find the optimal number of hidden Markov models to represent the usual motion patterns. These HMMs are then used for analyzing new crowd scenes and for detecting abnormal events.

In the context of natural and ecology studies, event detection is applied for the analysis of animal behaviour. As an example, in [13], a three-stage framework for event detection is described. It targets hunt detection in wildlife videos. The first layer of the framework extracts low-level description and motion information from the videos, while coping with camera motion. The second layer uses a neural network for the classification of the moving blobs, and it segments the input video clip into shots. The third layer applies user-defined event inference rules (e.g. a state machine) to verify whether a sequence of shots matches a target event. Another example is the work proposed in [25] which focuses on high-level events related to crowds of fish. It mainly addresses investigations on fish schooling characteristics. In particular, the proposed method exploits Lagrangian particle dynamics from fluid mechanics so as to consider the trajectories of fish as small particles in the fish flow.

2.2 Context

One of the most investigated context is video-surveillance, since it finds immediate use in security applications. The method proposed in [14] compares detected trajectories with a set of trajectories which are typical of intrusion patterns. It raises an alarm if a close match is found. A Gaussian mixture background model and a color-based blob tracker are used to detect and follow foreground objects. The tracker may generate more than one trajectory, because of occlusions or distinct movements of different body parts. Similar trajectories (in time and space) are merged into a single one. Then the comparison between an input trajectory and a database model is performed through a scale- and translation-invariant distance metric proposed by the authors. It also allows to quickly scan the database for possible matches. In [19], the authors describe an anomaly-detection system, with a use case for video-surveillance in a shop. Each trajectory is modeled by a HMM, and a distance matrix between all training trajectories is built. Multi-dimensional scaling (MDS) is applied to project trajectories onto a low-dimensional space. The projected vectors are then clustered using k-means. All cluster's trajectories are used to train a HMM representing the whole trajectory pattern. This method allows to both detect anomalous trajectories within the training set, and to perform online evaluation of new trajectories by computing the matching likelihood with the cluster HMMs.

Sports video clips have been one of the researchers' favourite contexts for event detection. Typically, the purpose of the existing methods is to detect salient parts of the videos (e.g. a team scoring a goal) for summarization. In some cases, this requires just a method to infer when something interesting is happening (e.g., through super-imposed graphics, camera movement, crowd views). In other cases, it is necessary to recognize the specific kinds of events of interest. In [15], the authors propose a sport video summarization system based on the detection of interesting "plays". Given a specific sport (case studies includes baseball, American football and Japanese sumo), a set of inference rules is defined to describe an interesting play in terms of sequence of scenes. For example, in baseball a play usually starts with a pitching scene, and if after a scene cut the camera is shooting the field, then the current play continues, otherwise the current play ends. Scene cuts are detected by comparing the colour histograms of two consecutive frames. Scene types are identified using features based on field colours and their spatial distribution, and the position of players and umpires. The rule inference matching scheme can be implemented by training a hidden Markov model with ground-truth play shots. In another work [16], the authors put particular focus on "field sports". They analyze videos in the search of features which may be an indication of interesting events, such as crowd images, audio activity, on-screen graphics, or scoreboard changes. A Support Vector Machine is then trained using the features computed from 210 events from different sports.

2.3 Event recognition approach

Given the different kinds of targets and contexts, it is extremely important to choose the best approach for describing events of interest and for matching such descriptions with the actual visual information contained in the video.

A common way to handle this task is to describe an event as a set of simple actions which can be easily recognized (e.g., an object moving in a certain direction or approaching another object, a set of speed variations). Such sets of simple actions can be recursive if necessary. In [17], the authors use trajectory data and a priori information on the scene to define three abstraction levels in the event recognition process: i) image features (e.g., size, speed, position, distance from reference objects), ii) mobile object properties (e.g., entering a certain area, approaching reference objects or other actors), and iii) scenarios (e.g., combinations of mobile object properties). Similarly, in [10] different scenarios are modeled with "basic properties" (e.g., trajectory, speed), states (e.g., a situation involving a set of actors at a certain time, or for a certain period) and events (variations of states).

Other works describe methods for event detection which aim at being as generic as possible to address a wider scope of application context. In such cases, little or no a priori

information is provided. The typical approach consists of using trajectory data (which can be represented in several ways, e.g. point sequence, histograms, hidden Markov models) and a clustering algorithm. It aims at identifying common motion patterns which can be associated to predefined events, or which can define new kinds of behaviour.

Porikli et al. [20] propose a method for the detection of unusual events based on spectral clustering. Histograms and HMMs use objects' characteristics such as speed, color, size, or aspect ratio. They serve as features for trajectory description. For each feature, an affinity matrix is built where the (i, j)th element shows how similar the ith and j th objects are, according to that feature. The affinity matrix are then decomposed using a certain number of the largest eigenvalues. After further transformations, a correlation matrix is computed. Clustering consists of grouping the elements whose results are highly correlated.

In [12], the authors apply a grammar rule induction framework to learn event rules. A clustering approach based on [26] is used to identify simple motion patterns. Hidden Markov models are trained to model each cluster, and are used as detectors of primitive events. A grammar induction algorithm is then applied to build the set of event rules. The induction algorithm evaluates grammar according to the Minimum Description Length principle [27].

Finally, in [28] the authors present a feature for event detection named Extended Relative Motion Histogram of Bag-of-visual-Words (ERMH-BoW). It aims at describing both the entities involved in an event and how they evolve. Instead of using raw motion distribution, which is noisy, the motion information of visual words is applied. Motion relativity histograms are adopted to handle problems caused by camera movements. Support Vector Machines using the ERMH-BoW descriptor are then trained to detect events in video clips.

To summarize, to our knowledge the literature does not contain event detection approaches working on animals in their natural environment. In fact, the mainstream approaches operate on controlled labs [29,30] with constant light conditions and high background-object contrast. Of course these conditions greatly simplify the task of mining the recorded videos. For human behaviour recognition, the most explored methods adopt visual concepts (as a direct representation of the scene) instead of using concepts that are more sensitive to viewpoint changes, such as trajectories and silhouettes [31]. This is not necessarily true in the cases of animals. Applications usually record animal and insect behaviour (both in controlled environments and in the natural environment) with rather fixed cameras whose viewpoints do not change often. Further, the structure of animal body (that varies more than human body) worsen the performance of visual concepts with respect to indirect scene representation (e.g., trajectories, body parts) [32].

3 The proposed system

The proposed system supports an event detection process organized in three main steps: (i) the detection, tracking and recognition of fish occurrences; (ii) the labeling of ground-truth video footage, on the basis of user-defined rules that retrieve potential event occurrences; (iii) the modeling of fish trajectories that allow our classification module to learn and detect the fish behaviours of interest. More specifically, the framework of our system is shown in Fig. 1, and involves the following stages:

1. The fish detection is carried out with an approach based on background modeling.
2. The modules for fish tracking and species recognition identifies the individual fish for each species of interest. These low-level results are stored in a database.
3. User-defined rules are specified through our web user interface, on the basis of the previously stored descriptions of fish occurrences. They describe the behaviours of interest, i.e., specific co-occurrences of fish from specific species.
4. A rule-based selection of potential events is performed. It provides a set of video excerpts that are potentially valid for a ground-truth dataset. The results are submitted to user validation and labeling, which are also performed through our UI.
5. The user-labeled fish trajectories are used as a training set for the trajectory classification module.
6. Finally, the trajectory classification module, the learnt behaviour models, and the user-defined rules are used to perform the event detection.

To understand better our event detection process, we report on a use case targeting the detection of pairing behaviours for the *Dascyllus Reticulatus*[3] species. The user-defined rule specified that this behaviour occurrence implies that two fish from this species co-occur within a specific timespan. The ground-truth trajectories were labeled using our UI. The pairing behaviour model for that species was learnt by our HMM-based trajectory classification module. Then, when a new video is processed, if the user-defined rule holds and if fish trajectories are classified as "pairing", then this specific event is detected and provided as output.

The video processing modules shown in the flowchart are described in detail in the following sections.

3.1 Fish detection

In the digital image processing domain, motion and object detection have drawn an important attention from machine

[3] http://www.fishbase.org/summary/Dascyllus-reticulatus.html.

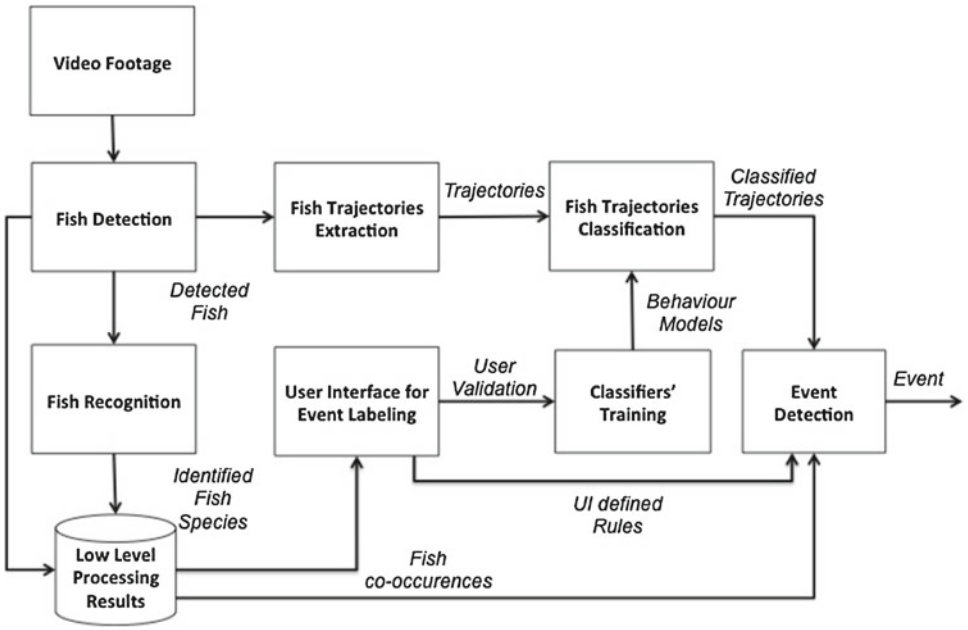


Fig. 1 Flow chart of the proposed event detection system

vision researchers. The most common approach to detect moving objects is based on modeling the background (i.e., the scene without objects of interest), and on assessing the difference between the frame under analysis and the modeled background. The mainstream approaches primarily deal with objects with a constrained motion (e.g., humans [33] or vehicles [34]). They are domain dependent, they mainly concern surveillance purposes and they cannot scale up to many different scenarios. Further, they demonstrated their limitations when dealing with noisy environments such as underwater environments [35].

Real-life underwater footage typically shows a combination of effects that make the task of object detection extremely difficult and challenging. First of all, the video quality is relatively low. This is due to technical difficulties in the communication between the underwater camera and the storage and processing servers, which limit the maximum network bandwidth. This limits the resolution and frame rate of the videos, thus causing a loss of details which could have improved the image processing. Secondly, the underwater scenes themselves are not easily modeled because of murky water, sudden lighting variations, background movement (e.g., algae), or periodic and multimodal background.

Porikli et al. [35] compared different detection algorithms under conditions that are similar to underwater conditions, but with humans as the main target. This comparative analysis shows that specific algorithms perform better under these conditions. These algorithms model the background with either (i) a mixture of probability density function (PDF) models [34,36,37]; (ii) a frequency transform to catch temporal color variation of background pixels [38]; or (iii) intrinsic images [39] given as the temporal median of the frames' reflectance component (which is assumed to be light invariant).

Recently, a simple and powerful approach in [40] models the background pixels with a set of neighborhood samples, instead of with an explicit pixel model. It was applied to different scenarios and showed promising results. But it performs well only when a limited number of effects occur in the scene. For instance, the Wave-Back algorithm [41] has good performance with repetitive scenes and with low-contrast colors, but not with erratic object movements and sudden lighting transitions. Regarding models based on a mixture of probability density function, they are able to model multimodal backgrounds, but they ignore the temporal correlation of color values.

To overcome the limitations of these detection approaches, we adopted Adaboost for its generalization capability [42]. The training process in Adaboost consists of building a binary classifier by using a set of weak classifiers:

$$C(X) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot c_t(X)\right) \tag{1}$$

where X is the training data, $c_t : X \rightarrow [0, 1]$ is a weak classifier and α_t is the weight of the classifier c_t so that $\sum_{t=1}^{T} \alpha_t = 1$. At each training step, Adaboost chooses the best classifiers, i.e. the ones that minimize the error criterion ϵ [42]:

$$\epsilon_t = \sum_i D_i \cdot e^{-y_i \cdot c_t(x_i)} \tag{2}$$

with D_i being the error distribution, and $y_i \in [0, 1]$ being the output of the classifier c_t at the ith iteration. According to

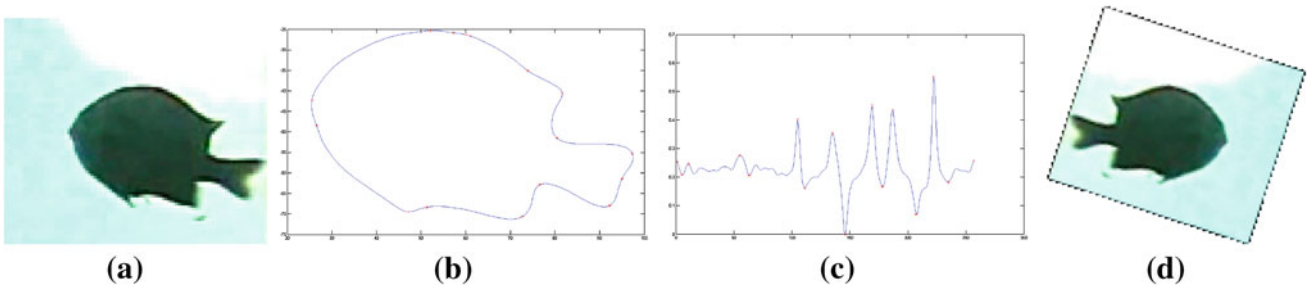


Fig. 2 Fish orientation demonstration: **a** original fish image; **b** fish boundary after gaussian filter; **c** curvature along fish boundary; **d** oriented fish image

the previous considerations, we used six background subtraction approaches as weak classifiers, namely: adaptive Gaussian mixture model (AGMM) [36], adaptive Poisson mixture model (APMM) [34], intrinsic model (IM) [39], Wave-Back (WB) [41], Codebook [43] and Vibe [40].

Moreover, to improve the overall performance of our fish detector, we added a side processing level which assesses the probability that a detected object is effectively a fish. We called that probability the *detection certainty*, as in the user interface of the Fig.RuleParam. To do so we exploit general and specific features of real-world objects [44]. In particular, we use "objectness" [45] and perceptual organization [46] to estimate general properties of real-world fish such as convexity, symmetry, well-defined boundary, visual contrast and cohesiveness. The features are given as input to a naive Bayes classifier, which is trained to distinguish objects of interest from false positives.

The detailed performance analysis of our fish detection system is given in Sect. 5.

3.2 Fish recognition

3.2.1 Feature extraction

Two pre-processing procedures are undertaken to improve the recognition rate. Firstly, the Grabcut algorithm [47] is employed to segment fish from the background, and to produce a binary mask. Secondly, we propose a preprocessing based on a streamline hypothesis to identify fish tails. We use the assumption that fish tails have an abrupt shape because fish need a frictional tail (caudal fin) to swim and to keep balance. To identify fish tails, we smooth the fish boundary with a Gaussian filter to eliminate noise, and we calculate the curvature of each boundary pixel following [48,49]:

$$\kappa(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2))^{\frac{3}{2}}} \tag{3}$$

where $X_u(u,\sigma)/X_{uu}(u,\sigma)$ and $Y_u(u,\sigma)/Y_{uu}(u,\sigma)$ are the first and the second derivative of $X(u,\sigma)$ and $Y(u,\sigma)$, respectively; $X(u,\sigma)$ and $Y(u,\sigma)$ are the convolution result of 1-D Gaussian kernel function $g(u,\sigma)$ with fish boundary coordinates $x(u)$ and $y(u)$. However, the pixel curvature is sensitive to local corners and we normalize it using the logarithm function:

$$\kappa_{\text{normalize}} = \begin{cases} log(\kappa) & \text{if } \kappa \geq 1 \\ -log(2-\kappa) & \text{if } \kappa < 1 \end{cases} \tag{4}$$

The fish boundary coordinates are weighted by their local curvature. The tail orientation is estimated by using the vector from the center of mask to the curvature-weighted center. A typical fish orientation procedure is illustrated in Fig. 2. Finally, every fish image is divided into four parts (head, tail, top, bottom) according to the relative positions from the fish center.

This method achieved a stable accuracy (95 %) when identifying the tail side in 1,000 hand-labeled images. This curvature orientation method selects the relative curvature center which is invariant to contour scale changes.

After this preprocessing, 66 types of feature are extracted. They are a combination of color, shape and texture properties of the four parts of the fish (tail, head, top, bottom) and of the whole fish. We use normalized color histogram in the Red&Green channel and the Hue component in HSV color space. These color features are normalized to minimize the effect of illumination changes. We recompute the range of every bin according to the average distribution over all samples, and map them into a 11-bin histogram to take full advantage of all bins, as shown below:

$$\widetilde{B}_i = \sum_{j=a_i}^{a_{i+1}} B_j \quad s.t. \quad a_i = \min\left\{ X \in \mathbb{N}^+ \| \Sigma_{j=1}^{X} \overline{B}_j \geq \frac{i}{11} \right\} \tag{5}$$

where $B_j, j \in \{1,\ldots,50\}$ is the original color histogram bin, $\overline{B}_j, j \in \{1,\ldots,50\}$ is the averaged histogram over all samples and $\widetilde{B}_i, i \in \{1,\ldots,11\}$ is the recomputed bin.

To describe the fish texture, we calculate the co-occurrence matrix, the Fourier descriptor and the Gabor filter. The grey level co-occurrence matrices describe the co-occurrence frequency of two grey scale pixels at a given distance d [50]:

$$C_{\Delta u,\Delta v}(i,j) = \sum_{p=1}^{n}\sum_{q=1}^{m}\begin{cases}1 & \text{if } I(p,q)=i \text{ and } I(p+\Delta u, q+\Delta v)=j\\ 0, & \text{otherwise}\end{cases} \quad (6)$$

The frequency is calculated for several orientations λ. We compute contrast, correlation, energy, entropy, homogeneity, variance, inverse difference moment, cluster shade, cluster prominence, max probability, auto correlation, dissimilarity. These 12 features are useful as they are the first selected features by the feature selection procedure. Histogram of oriented gradients and moment Invariants, as well as Affine Moment Invariants, are employed as the shape features. Furthermore, some specific features like tail/head area ratio, tail/body area ratio, etc. are also included. All features are z-score normalized by subtracting the mean, and divided by the standard deviation.

3.2.2 Classification

We use a hierarchical classification tree based on SVM, which achieves better performance on similar classes. First, a one-vs-one strategy with a voting mechanism is introduced to convert the binary SVM into a multi-class classifier [51]. Each class is trained in a set of binary classifiers against each other class. A sequential forward feature selection algorithm is applied by each classifier to select the best subset of discriminative features at that node in the hierarchy. Finally, the hierarchical classifier is a taxonomy tree, constructed according to the fish species taxonomy. This tree is pre-defined. It reflects the homologous similarity between species as defined by the biologists. Considering the species currently addressed, this tree splits all classes into five groups at the first level according to their family synapomorphy characteristics. It leaves a few similar species to deeper layers in the tree, where a customized classifier is applicable.

3.3 Fish tracking for trajectory extraction

After fish detection and recognition, the next step consists of following them in a video across consecutive frames. This task is commonly called "object tracking". A tracking algorithm aims at recognizing that two regions in two different frames represent the same fish. This comparison is performed by: (i) motion analysis, e.g., two occurrences of a single fish in a two consecutive frames must be consistent with their position, speed and direction; and (ii) appearance analysis, e.g., features like shape, colour, size, or textures must be similar for two occurrences of a single fish in a two consecutive frames. In the underwater environment, this task proves to be much harder than in typical tracking applications involving humans. Beyond all the aspects that may affect fish detection performance (e.g., murky water, sudden lighting variations or background movement), the main challenge of fish tracking lies in the nature of the targets themselves, i.e., fish. Their appearance is subject to sensible changes across a video, due to the flexibility of fish bodies and to changes in light conditions. Their typical erratic motion makes their direction less predictable, especially in videos with low frame rate. Moreover, the partial or total overlap of two fish (occlusions) is very frequent. It requires the tracker to cope with the temporary "loss" of a fish and the re-identification when the fish appears again.

To tackle these problems, we developed an algorithm specifically designed for tracking fish in unconstrained underwater environments [6,9]. It is based on a covariance representation of fish features [52]. This approach models fish as the covariance matrices of a set of features that uses each pixel belonging to the object's region. This representation allows to embody both the spatial and statistical properties. Unlikely, histogram representations disregard the structural arrangement of pixels, and appearance models ignore statistical properties.

In detail, for each fish detected in a frame, the corresponding covariance matrix is computed by the following procedure. First a feature vector is built for each pixel. It consist of the pixel coordinates, the RGB and hue values and the mean and standard deviation of the histogram of a 5×5 window containing the target pixel. Then the covariance matrix, modeling the fish, is computed from this feature vector and associated to the detected fish. Afterwards the covariance matrix is used to compare the currently tracked fish, in order to decide which ones resemble the most.

The achieved results [6] show that the proposed algorithm can accurately track a fish even when it is temporarily hidden or when similar fish are present in the scene. However, the accuracy of the algorithm is strongly linked to the accuracy of the detection algorithm. This is because the fish tracking assumes that all and only moving objects are provided by the fish detection and its underlying motion algorithm. For this reason, tracking may fail because of detection inaccuracy.

3.4 Fish trajectory classification

Candidate events can be preliminarily detected by user-defined rules. For instance, rules can retrieve isolated fish occurrences (solitary behaviour), co-occurrences of fish from the same species (pairing or schooling behaviour) or from different species (feeding behaviour). However, in most cases rule criteria are not enough to reliably detect the presence of an event. For this reason, we couple the rule-based detection system with a trajectory classifier. This aims at filtering out false positives amongst event candidates provided by the rules.

We based our approach on the idea that, for a given type of event, it is possible to capture a common motion pattern in the trajectories of the fish involved in these particular events. This especially apply to events involving fish to fish interactions (e.g., solitary or pairing behaviours). In the training stage, the common motion patterns are extracted from the ground-truth events labeled by users (see Sect. 4). They are then are encoded into the classifier. Finally, in the event detection phase, the motion patterns are used to verify whether the candidate trajectories match the learnt patterns or not.

To choose the type of classifier that suits our purposes, a central problem is to find an appropriate form of trajectory representation. The typical point-sequence representation contains all the information describing the movement of an object, but is often difficult to work with. This is because the comparison of trajectories with varying length requires a normalization of the number of points, with the risk of over- or under-sampling. Moreover, it is difficult to represent a generic motion pattern as a sequence of points. Histograms of features such as position, speed, orientation (e.g. [20]) could also be employed to describe trajectories. But it loses all temporal information, which is an essential part of the pattern recognition process. On the contrary, hidden Markov models (HMMs) can intrinsically encode spatio-temporal sequences of data. They also support intuitive algorithms to generate sample trajectories and to check if an input trajectory matches the learnt pattern. They are often used in the description of trajectories and trajectory clusters and patterns.

A hidden Markov model is a stochastic model describing a Markovian process where the states are not directly observable, contrary to a regular Markov chain. The estimation of the current state is then performed by analyzing the systems 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 applicable to temporal pattern recognition [53]. 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. The structure and the dimensions of these matrices can vary if there are multiple output variables, or if the distribution is continuous as it is the case for this work. A description of continuous-output HMMs using mixtures of Gaussians is presented in [53].

In this context, we extend each trajectory $T = \{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}$ (i.e. the sequence of centroid coordinates provided by the tracking algorithm) to a vector $T_{ext} = \{(x_0, y_0, v_0, \theta_0), (x_1, y_1, v_1, \theta_1), \ldots, (x_n, y_n, v_n, \theta_n)\}$, where v_i and θ_i represent, respectively, the speed and the orientation variations with respect to the previous point.

During the training stage, for each fish species and for each event type that we take into consideration, all corresponding ground-truth trajectories are used to train a hidden Markov model. Our HMMs are then used to verify whether a given trajectory matches a specific event.

Once all classifiers have been trained, they need to be integrated into the event detection framework. The integration is performed in two separate ways, depending on the kind of events:

- Solitary event: given the trajectory of a fish which appears to be spatially isolated from the other ones (according to a user-defined rule), a solitary behaviour event is detected if the likelihood that the trajectory would be generated by the "solitary" HMM for that species is higher than a specified threshold.
- Pairing event: given the trajectories of two fish from the same species which appear to be following each other for a long enough time (according to a user-defined rule), a pairing behaviour event is detected if the likelihood that both trajectories would be generated by the "pairing" HMM for that species is higher than a specified threshold.

These procedures allow to integrate the HMM classifiers with the rule-based system that selects trajectories representing possible events. In Sect. 5 we show the parameters used in the training and testing phases, and we evaluate the system performance w.r.t. the ground-truth events.

3.5 Event detection

The event detection modules operate on the outputs of the previous modules. It employs simple rules combining the user-defined rules (defined during the event labeling process) and the result of trajectory classification. We currently implemented the detection of solitary end pairing events, with rules of the following form:

- Solitary events: IF a fish is of species X and does not co-occur with any fish within a timespan of T and during at least F frames, and IF its trajectory is classified as solitary THEN a Solitary Event for Fish Species X is identified.

Table 1 Example of rules used for event detection

Fish species	Behaviour	Rule
Chromis margaritifer	Solitary	B:solo, N:1, S:2, F:30, T:35
Chaetodon trifascialis	Solitary	B:solo, N:1, S:6, F:10, T:10
Scolopsis bilineata	Solitary	B:solo, N:1, S:8, F:25, T:25
Dascyllus reticulatus	Pairing	B:pair, N:2, S:1, F:10, T:25
Plectrogly-phidodon dickii	Pairing	B:pair, N:2, S:3, F:5, T:20
Pomacentrus moluccensis	Pairing	B:pair, N:2, S:5, F:10, T:5

– Pairing events: IF two fish are of species X and co-occur within a timespan of T and during at least F frames containing exactly two fish, and IF both trajectories are classified as pairing THEN a Pairing Event for Fish Species X is identified.

For a given event, several rules can be defined by users (or by the same user). In this case, we select the rule with the highest number of labeled video excerpts. Table 1 shows some of the rules used for event detection, where B stands for behaviour type, N for the number of co-occurrences of fish from the same species, S for fish species, F for the number of frames in which fish co-occur, and T for timespan.

4 Intelligent user interface for video labeling

The collection of training datasets is a tedious and time-consuming task. It involves filtering, browsing and watching numerous videos. Further, identifying meaningful events necessitates an understanding of the domain-specific interests of end-users. For instance, groups of fish can gather for reproduction activities or for feeding activities, depending on the species.

Our user interface addresses both of these concerns, i.e., (i) reducing the effort needed to collect training datasets, and (ii) handling the specification of meaningful events.

We based the specification of meaningful events on the user study conducted for the Fish4Knowledge project.[4] End-users expressed interest in fish interactions related to demographics, reproduction, feeding, and environmental conditions. They elicited ten species that are the most interesting to study because their behaviours are representative of the ecosystem conditions. We derived the specific fish behaviours of interests on the basis of descriptions of the ten species provided by end-users and by the FishBase project.[5] In this paper we investigate pairing and solitary behaviours, as they address biologists' interests in demographics, reproduction, feeding, and environmental conditions. The meaning of pairing and solitary events depend on the species involved, and the Table 2 summarizes their interpretation.

Table 2 Interpretation of solitary and pairing events depending on fish species

Fish species	Solitary	Pairing
Dascyllus reticulatus	Abnormal	Breeding
Chromis margaritifer	Normal	Breeding
Plectrogly-phidodon dickii	*Unknown*	Breeding
Acanthurus nigrofuscus	Abnormal	*Unknown*
Pomacentrus moluccensis	Abnormal	Breeding
Chaetodon trifascialis	Normal	Normal breeding
Zebrasoma scopas	Juvenile	Rare
8 Scolopsis bilineate	Juvenile	Adult
Amphiprion clarkii	*Unknown*	Breeding
Siganus fuscescens	Abnormal	*Unknown*

To reduce the effort needed for collecting training datasets, we designed a rule-based interface. It helps targeting meaningful events by supporting user-defined specification of fish co-occurrences to retrieve. Users can define the rule parameters that target specific species, number of fish, delay between fish and duration of co-occurrences. They can also apply specific sampling methods by randomizing the ordering of the retrieved samples, by selecting the time periods to sample, and by specifying the number of samples needed. In particular, the UI achieves the following points:

– The effort needed to define the rule parameters is reduced to a limited number of form inputs to fill in, and user inputs are integrated in human-understandable sentences.
– The rule supports sufficient flexibility to address the set of events of interests from Table 2.

The user interface functionalities support (i) the retrieval of video excerpts that display the co-occurrences of interest, and (ii) the manual selection of video excerpts that are suitable for the training dataset. It organizes the dataset collection task in three steps:

1. Define the rule, and the sampling method.
   Users are supported with two simple rules, and a set of parameters they can modify. The most important rule supports the retrieval of solitary fish and pairing fish. It covers most of the events of interest from Table 2. An additional rule can be used to retrieve co-occurrences of fish from two specific species. For instance, this rule can be used to analyze the interactions of juvenile Acanthurus

[4] http://homepages.inf.ed.ac.uk/rbf/Fish4Knowledge/DELIVERABLES/Del21.pdf.

[5] http://fishbase.org.

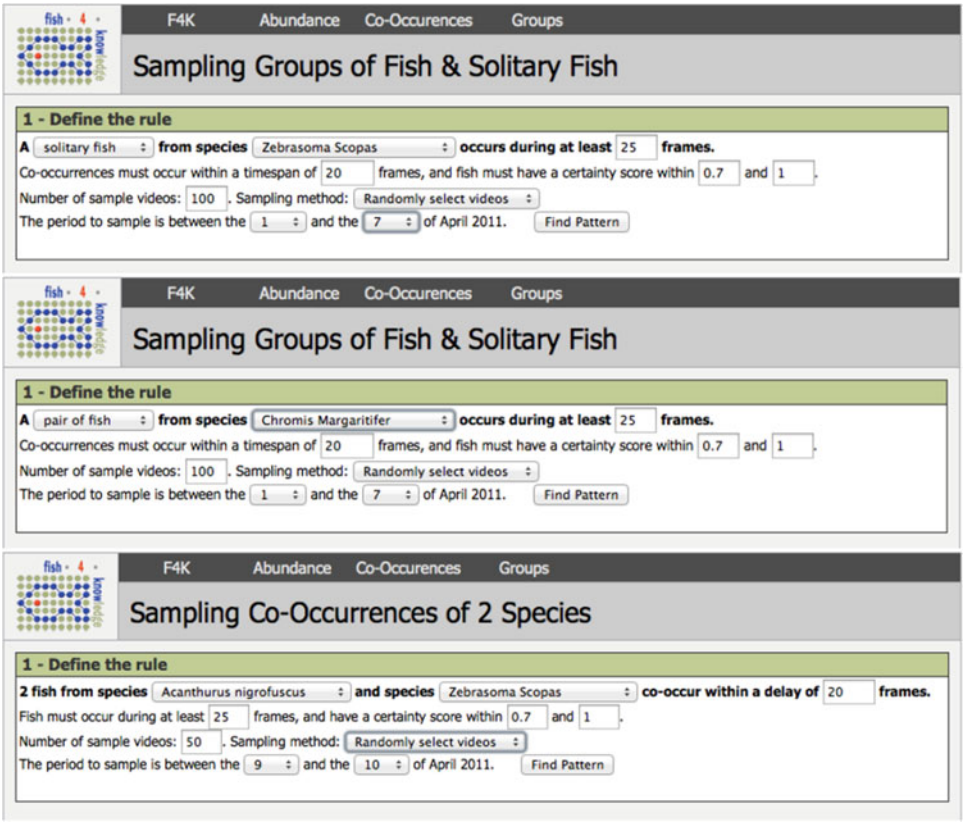


Fig. 3 Screenshots of user-defined rules for retrieving solitary and paring fish (first two images), and for retrieving co-occurrences of two species (last image)

Nigrofuscus with other species. Figure 3 shows how our user interface supports the specification of rule parameters.

2. Manually select valid video samples.
   Users are provided with a list of video samples that satisfy the rule they defined. Users can watch the video samples. If a sample is a good example of the event of interest, users can click on the sample to include it in the training dataset. The Fig. 4 shows a selected and a discarded video sample in our user interface.
3. Store the training dataset.
   After selecting a set of training video samples, users can label the training dataset and describe what event detection it supports. The Fig. 5 gives an example of a label for a training dataset. When storing the dataset, the system saves the rule parameters and all the video samples it retrieved: the manually selected samples, flagged as valid samples, and the discarded samples.

5 Experimental results

The proposed system consists of different modules integrated together. Therefore the overall performance depends on the performance of each low level processing module.

5.1 Fish detection and tracking

For the evaluation of the detection and tracking modules, we used eight videos (of ten minutes each) of the Fish4Knowledge repository. The videos had resolutions of 320×240, a 24-bit color depth, and a frame rate of 5 fps. The videos were selected for their specific features that allow to test the effectiveness of the fish detection under the following conditions: multimodal background, sudden illumination changes, high water turbidity, background objects, low contrast and camouflage phenomena.

The ground-truth was manually labeled using the tool described in [54] and contained a total of 31,221 detections, corresponding to 2,113 different fish (a fish may have more than one detection). The performance evaluation of the detection algorithms was carried out both at object level, to test the effectiveness of fish objects detection, and at pixel level, to test their capabilities to preserve objects' shape. As described in Sect. 3.2, we used six different object detection algorithms, namely: adaptive Gaussian mixture model (AGMM) [36], adaptive Poisson mixture model (APMM) [34], intrinsic model (IM) [39], Wave-Back (WB) [41], Codebook [43] and Vibe [40]. They were then used as weak classifiers in the Adaboost approach. The achieved performance, both at object level and at pixel level, are reported in terms of ROC

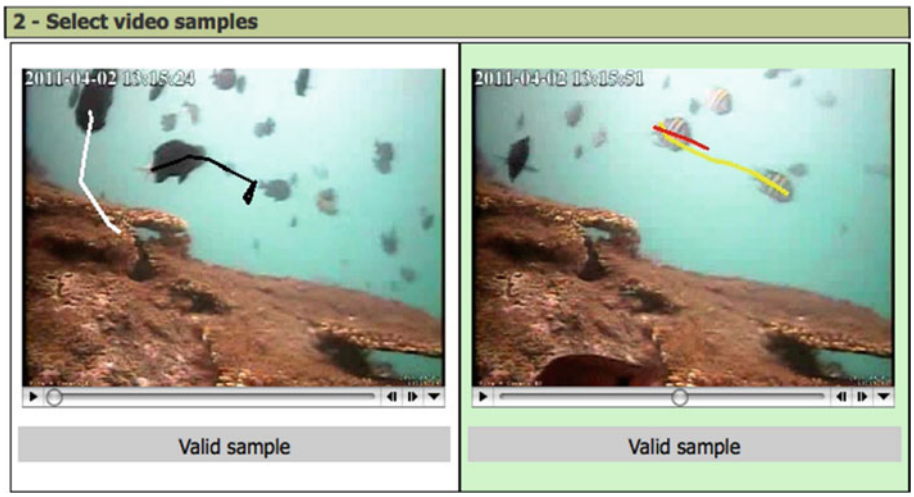


Fig. 4 Users can select valid video samples (e.g., the video on the right is selected) and discard the others

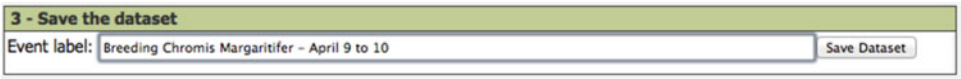


Fig. 5 Users can label the training dataset to describe the targeted event

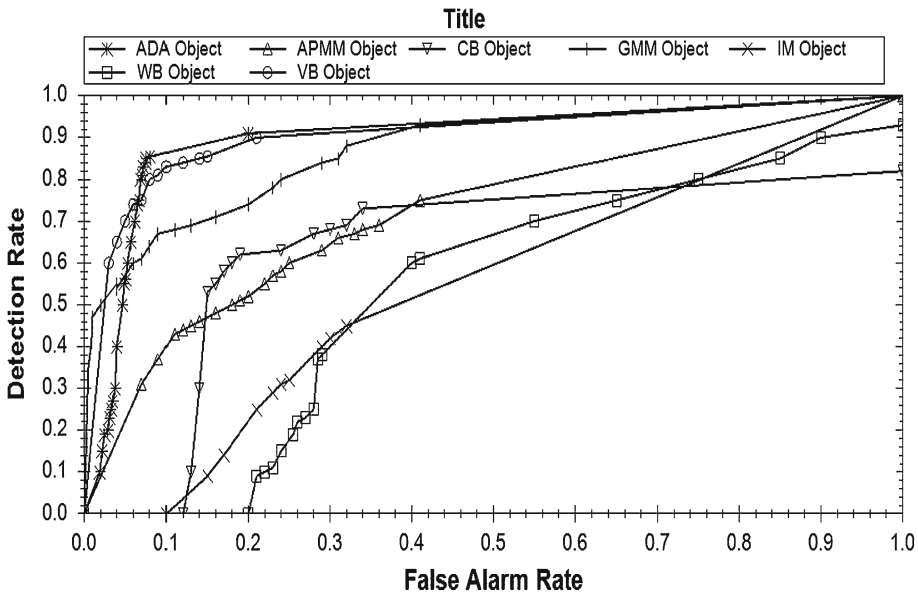


Fig. 6 ROC curves for the object level performance of the adaptive Gaussian mixture model (AGMM), adaptive Poisson mixture model (APMM), intrinsic model (IM), wave-back (WB), Codebook and Vibe, and Adaboost approaches

curves in Figs. 6 and 7. In these figures, the x axis represents the false alarm rate (FAR), and the y axis represents the detection rate (DR), defined as:

$$\mathrm{DR} = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{7}$$

$$FAR = \frac{N_{FP}}{N_{TP} + N_{FP}} \tag{8}$$

where N_{TP}, N_{FP} and N_{FN} are, respectively, the number of true positives, false positives and false negatives. For the evaluation at the object-level, these values refer to the whole objects (divided in foreground and background objects), whereas at a pixel-level they refer to the pixels belonging to a fish and the pixels belonging to the background.

The same ground-truth dataset was used to test the tracking performance. The performance of our algorithm was compared with the ones achieved, respectively, by the CONDENSATION [55] (based on particle filter), and by the CAMSHIFT (which was previously tested on underwater domain [4]). To assess the *ground-truth-vs-algorithm* comparison, we adopted the following metrics, defined in [6]. They describe the performance of a tracking algorithm both globally, i.e., at the trajectory level, and locally, i.e., at the single tracking decision level.

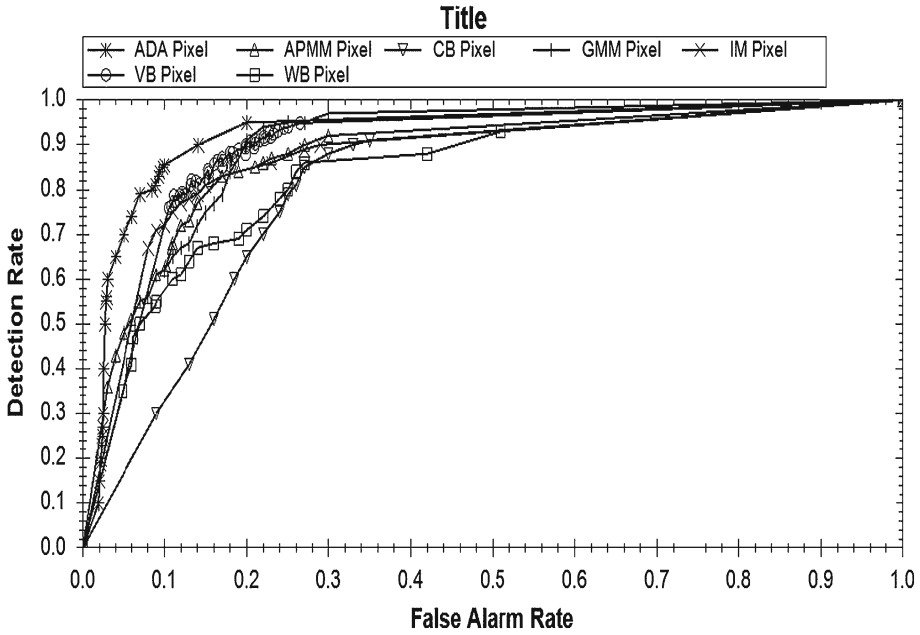


Fig. 7 ROC curves for the pixel level performance of the adaptive Gaussian mixture model (AGMM), adaptive Poisson mixture model (APMM), intrinsic model (IM), wave-back (WB), Codebook and Vibe, and the Adaboost approach

- *Correct counting rate (CCR)* percentage of correctly identified fish out of the total number of ground-truth fish.
- *Average trajectory matching (ATM)* average percentage of common points between each ground-truth trajectory and its best-matching tracker-computed trajectory.
- *Correct decision rate (CDR)* let a "tracking decision" be an association between a fish at frame t_1 and a fish at frame t_2, where $t_1 < t_2$; such tracking decision is correct if it corresponds to the actual association, as provided by the ground-truth. The correct decision rate is the percentage of correct tracking decisions, and gives an indication on how well the algorithm performs in following an object, which is not necessarily implied by the average trajectory matching (see Fig. 8).

Table 3 shows the results obtained by the covariance tracking algorithm compared to the ones achieved by CAMSHIFT and CONDENSATION, using the above-described metrics. Our tracking approach outperforms two of the most common and powerful state-of-the-art approaches. It shows a high absolute accuracy, being able to correctly identify more than 90 % of unique objects with a very high degree of correspondence to the ground-truth trajectories.

5.2 Fish recognition

The fish recognition modules were tested on 3,179 fish images. We used more images than for the detection of ground-truth, as we needed a sufficient number of images for each fish species. We used a sixfold cross validation procedure. The training and testing sets were isolated, so that

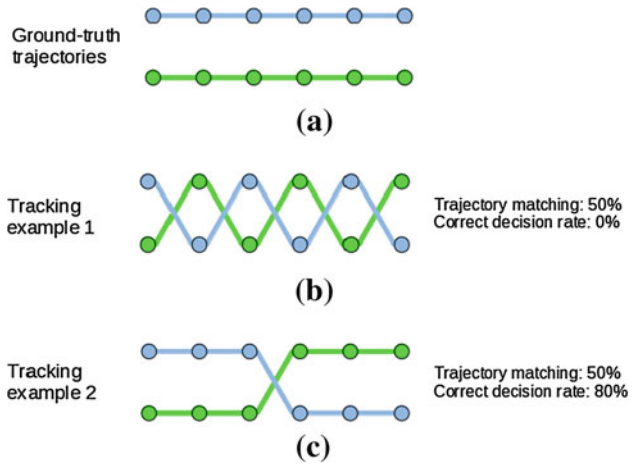


Fig. 8 The difference between the trajectory matching score and the correct decision rate. **a** Shows two ground-truth trajectories of two fish, whereas the other two images represent two examples of tracking output. **b** Although the tracker fails at each tracking decision the trajectory matching score is 50 %, whereas the correct decision rate is 0. Contrarily, in **c** the tracker fails only in one step and the trajectory matching score is 50 % (as the previous case) whereas the correct decision rate is 80 % (4 correct associations out of 5)

fish images from the same trajectory sequence were not used during both training and testing. Sequential forward feature selection was applied at each node. Then we trained a customized classifier at each node for specific classes. For the hierarchical tree, we used the prior knowledge of the fish taxonomy system (shown in Fig. 9) which has a similar tree structure. This tree splits all classes into five groups at the first level, according to their family synapomorphies. It leaves a few similar species to deeper layers, where a customized classifier is applicable.

Table 3 Tracking performance comparison

	Covariance tracker (%)	CAMSHIFT (%)	CONDENSATION (%)
CCR	91.3	83.0	89.2
ATM	95.0	88.2	91.4
CDR	96.7	91.7	94.3

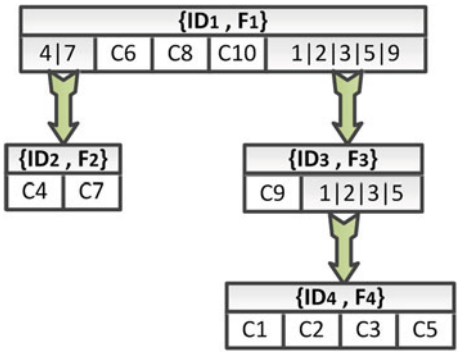


Fig. 9 Taxonomy tree

Table 4 Fish recognition result

Algorithm	Average accuracy (%)
Flat SVM	86.32
Taxonomy tree	90.30

Table 5 Recognition results in terms of average recall for fish-species

Fish species	Average recall	Standard deviation
Dascyllus reticulatus	0.974	0.013
Chromis margaritifer	0.939	0.051
Plectrogly-phidodon dickii	0.945	0.047
Acanthurus nigrofuscus	0.761	0.068
Pomacentrus moluccenis	0.974	0.028
Chaetodon trifascialis	0.985	0.037
Zebrasoma scopas	0.553	0.329
Scolopsis bilineate	0.964	0.087
Amphiprion clarkii	0.933	0.163
Siganus fuscescens	1	0

We compared the hierarchical classification (average recall: 90.30 %) against the flat SVM classifier (average recall: 86.32 %). The taxonomy tree controls the maximum depth and keeps balanced (Table 4).

Table 5 shows the average recall for each fish species, over six cross validations.

Table 6 Ground-truth trajectories for each fish species

ID	Fish species	Behaviour	Trajectories
DR_S	*Dascyllus reticulatus*	Solitary	104
CM_S	*Chromis margaritifer*	Solitary	106
PD_S	*Plectrogly-phidodon dickii*	Solitary	95
PM_S	*Pomacentrus moluccenis*	Solitary	60
CT_S	*Chaetodon trifascialis*	Solitary	57
SB_S	*Scolopsis bilineate*	Solitary	237
AC_S	*Amphiprion clarkii*	Solitary	63
SF_S	*Siganus fuscescens*	Solitary	51
DR_P	*Dascyllus reticulatus*	Pairing	104
CM_P	*Chromis margaritifer*	Pairing	144
PD_P	*Plectrogly-phidodon dickii*	Pairing	138
CT_S	*Chaetodon trifascialis*	Pairing	90
SB_P	*Scolopsis bilineate*	Pairing	104

5.3 Fish trajectory classification

For each event type and fish species shown in Table 6, we trained a hidden Markov model specialized in the recognition of the trajectory patterns. Each HMM was trained using the Baum–Welch algorithm, and the number of states and output mixtures were both set to four.

Table 6 shows the number of ground-truth trajectories labeled for each of the considered events of interest. Interestingly, we did not collect a ground-truth for all fish species and for all behaviour types. This is either because some behaviours were not significant for marine biologists, because we did not detect and recognize any fish of some specific species, or because the number of detections was not sufficient to train HMM.

For each HMM, 70 % of the corresponding events were used for training and 30 % for testing. In total the trajectories classification module was trained on 947 trajectories and tested on the remaining 406 trajectories. Table 7 shows the classification performance of each single HMM, in terms of detection rate (DR) and false alarm rate (FAR) given in percentage.

Interestingly, our HMM-based trajectory classification module reached on average a DR of about 80 %, and a FAR of 24 %. In some cases, the number of false positives was relevant (e.g for DR_S about 35 %), but they were then reduced when the trajectory classification was integrated with the user-defined rules.

5.4 Event detection

Our trajectory-based event detection system was trained on 1,068 events and tested on 499 events, divided in 320 events of interest and 179 events of no interest. The events of interest

Table 7 Performance of trajectory classification, by species and event type

ID	DR (%)	FAR (%)
DR_S	70.9	35.7
CM_S	71.8	39.1
PD_S	72.4	33.3
PM_S	100.0	33.3
CT_S	100.0	33.3
SB_S	77.4	41.5
AC_S	73.6	0.0
SF_S	100.0	0.0
DR_P	75.0	27.7
CM_P	73.9	30.7
PD_P	75.0	14.2
CT_S	71.4	25.0
SB_P	73.3	33.3
Average	81.9	24.11

are the ones shown in Table 6. The performance evaluation of the event detection was assessed using normalized detection cost (NDC) [1,56], defined as a weighted linear combination of missed detection (MD) and false alarm (FA) probabilities. The NDC for a specific event is given by:

$$\mathrm{NDC} = C_{\mathrm{MD}} \cdot P_{\mathrm{MD}} \cdot P_{\mathrm{T}} + C_{\mathrm{FA}} \cdot P_{\mathrm{FA}} \cdot (1 - P_{\mathrm{T}}) \qquad (9)$$

with $P_{\mathrm{MD}} = \frac{N_{\mathrm{MD}}}{N_{\mathrm{T}}}$, and $P_{\mathrm{FA}} = \frac{N_{\mathrm{FA}}}{N_{\mathrm{T}}}$ that are, respectively, the missed detection and false alarm probabilities. N_{E}, N_{T} , N_{MD}, N_{FA} are, respectively, the number of the specific event instances, the total numbers of events, missed detections and false alarms. P_{T} is the a priori rate of event instances E. C_{MD} and C_{FA} are, respectively, the costs of MD and FA. We set C_{MD} and C_{FA}, respectively, to 10 and 15 to keep false alarms and missed detections balanced, as a high number of false alarms might affect fish behaviour analysis, but at the same time we do not want to miss important events. The NDC was computed for all the species-related events of Table 6, and the results are reported in Table 8.

The achieved results highlight how our system performs quite well in detecting fish behaviour events. These results show that the system performance is comparable to those of by state-of-the-art approaches performing on much simpler events [1,9].

6 Concluding remarks

Understanding fish behaviour is of key importance for marine biologists that study the underwater environment and the related climate conditions. However, in the recent past the investigation of the marine ecosystems was partial and limited because of the difficulty of collecting useful data. In particular the mainstream techniques influence the environment under observation (e.g., the intrusion of divers).

In this context, the use of embedded cameras simplifies the collection of relevant data for studying fish populations and behaviours, while limiting the intrusive effects. But it is not feasible for a human operator to manually analyze the enormous amount of recorded data. To address both the needs for (i) an automatization of the video processing, and for (ii) a reduction of the effort involved in manual inspection of video, we propose in this paper a rule-based event detection system able to identify specific behaviours of the most common species in the Taiwanese coral reef.

Table 8 Evaluation results for the events shown in Table 6

E	N_E	N_T	MD	FA	P_{MD}	P_{FA}	P_{T}	NDC
DR_S	31	499	9	5	0.018	0.010	0.062	0.152
CM_S	32	499	9	6	0.018	0.012	0.064	0.180
PD_S	29	499	8	2	0.016	0.004	0.058	0.066
PM_S	18	499	0	1	0	0.002	0.036	0.029
CT_S	17	499	0	0	0	0	0.034	0
SB_S	71	499	16	24	0.032	0.048	0.142	0.663
AC_S	19	499	5	0	0.010	0	0.038	0.004
SF_S	15	499	0	0	0	0	0.030	0
DR_P	16	499	4	3	0.008	0.006	0.032	0.090
CM_P	23	499	6	5	0.012	0.010	0.046	0.149
PD_P	20	499	5	0	0.010	0	0.040	0.004
CT_S	14	499	4	1	0.008	0.002	0.028	0.031
SB_P	15	499	4	0	0.008	0	0.030	0.002

To the best of our knowledge, our event detection system represents one of the first attempts in recognizing animal behaviours in the underwater domain. It also contributes to the event detection research field by providing information on which approach and algorithm might work better in crowded and complex domains such as the one we are dealing with. The achieved results for the fish detection, tracking, recognition and trajectory classification and event detection are promising, especially considering the difficulties of underwater environment. Although the system fails in some cases, it allows marine biologists to study fish behaviour more reliably and faster than with the approaches traditionally employed in marine biology.

At the moment we detect only two types of behaviour for the most seen species. As future work, we plan to extend the proposed approach to social behaviours and interactions between fish (e.g., feeding, predator-prey, territorial, reproduction, nursing). We also target events involving more than two fish, e.g. schools of fish, and fish-background interactions (e.g., biting on coral). The case of the interaction between fish involves the effective modeling a set of trajectories, as opposed to the approach proposed here where trajectories are considered individually (i.e. not influenced by other fish). And the case of fish-background interactions involves the description and identification of background regions through a powerful scene segmentation approach.

References

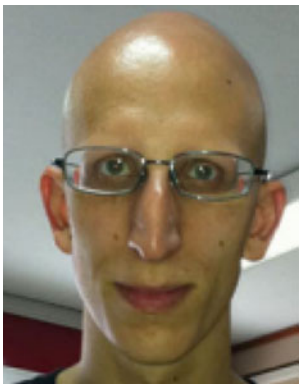
1. Gkalelis, N., Mezaris, V., Kompatsiaris, I.: High-level event detection in video exploiting discriminant concepts. In: 9th International Workshop on Content-Based Multimedia Indexing (CBMI 2011). Madrid, Spain, 06/2011 (2011)
2. Liao, M.-Y., Chen, D.-Y., Sua, C.-W., Tyan, H.-R.: Real-time event detection and its application to surveillance systems. In: Proceedings of 2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006, vol. 4, p. 512 (2006)
3. Ballan, L., Bertini, M., Bimbo, A.D., Seidenari, L., Serra, G.: Event detection and recognition for semantic annotation of video. Multimedia Tools Appl. **51**, 279–302 (2011)
4. Spampinato, C., Chen-Burger, Y.-H., Nadarajan, G., Fisher, R.: Detecting, tracking and counting fish in low quality unconstrained underwater videos. In: Proceedings of 3rd International Conference on Computer Vision Theory and Applications (VISAPP), vol. 2, pp. 514–519 (2008)
5. Spampinato, C., Giordano, D., Di Salvo, R., Chen-Burger, Y.-H.J., Fisher, R.B., Nadarajan, G.: Automatic fish classification for underwater species behavior understanding. In: Proceedings of the first ACM international workshop on analysis and retrieval of tracked events and motion in imagery streams, pp. 45–50. ARTEMIS '10, ACM, New York, NY, USA (2010)
6. Spampinato, C., Palazzo, S., Giordano, D., Kavasidis, I., Lin, F.-P., Lin, Y.-T.: Covariance based fish tracking in real-life underwater environment. In: VISAPP (2), pp. 409–414 (2012)
7. Rijnsdorp, A.D., Peck, M.A., Engelhard, G.H., Mšllmann, C., Pinnegar, J.K.: Resolving the effect of climate change on fish populations. ICES Journal of Marine Science: Journal du Conseil **66**(7), 1570–1583 (2009)
8. Scott, G.R., Sloman, K.A.: The effects of environmental pollutants on complex fish behaviour: integrating behavioural and physiological indicators of toxicity. Aquatic Toxicol **68**(4), 369–392 (2004)
9. Spampinato, C., Palazzo, S., Boom, B., van Ossenbruggen, J., Kavasidis, I., Di Salvo, R., Lin, F.-P., Giordano, D., Hardman, L., Fisher, R.: Understanding fish behavior during typhoon events in real-life underwater environments. Multimedia Tools Appl. pp. 1–38 (2012). doi:10.1007/s11042-012-1101-5
10. Cupillard, F., Avanzi, A., Bremond, F., Thonnat, M.: Video understanding for metro surveillance. In: IEEE International Conference on Networking Sensing and Control, vol. 1, pp. 186–191, IEEE (2004)
11. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: IEEE 11th International Conference on Computer Vision, vol. 23, pp. 1–8 (2007)
12. Zhang, Z., Huang, K., Tan, T., Wang, L.: Trajectory series analysis based event rule induction for visual surveillance. In: IEEE Conference on Computer Vision and, Pattern Recognition, pp. 1–8 (2007)
13. Haering, N., Qian, R.J., Sezan, M,I.: A semantic event-detection approach and its application to detecting hunts in wildlife video (2000)
14. Liao, M.-Y., Chen, D.-Y., Sua, C.-W., Tyan, H.-R.: Real-time event detection and its application to surveillance systems. In: Proceedings of the IEEE International Symposium on Circuits and Systems (2006)
15. Li, B., Ibrahim Sezan, M.: Event detection and summarization in sports video. In: Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries CBAIVL 2001, pp. 132–138 (2001)
16. Sadlier, D.A., O'Connor, N.E.: Event detection in field sports video using audio-visual features and a support vector Machine (2005)
17. Medioni, G., Cohen, I., Bremond, F., Hongeng, S., Nevatia, R.: Event detection and analysis from video streams. IEEE Trans. Pattern Anal. Mach. Intell. **23**(8), 873–889 (2001)
18. Assfalg, J., Bertini, M., Colombo, C., Bimbo, A.D., Nunziati, W.: Highlight extraction in soccer videos (2003)
19. Suzuki, N., Hirasawa, K., Tanaka, K., Kobayashi, Y., Sato, Y., Fujino, Y.: Learning motion patterns and anomaly detection by Human trajectory analysis. In: IEEE International Conference on Systems, Man and, Cybernetics, pp. 498–503 (2007)
20. Porikli, F., Haga, T.: Event detection by eigenvector decomposition using object and frame features. In: Conference on Computer Vision and Pattern Recognition, Workshop (2004)
21. Huang, C.-L., Shih, H.-C., Chao, C.-Y.: Semantic analysis of soccer video using dynamic Bayesian network (2006)
22. Piciarelli, C., Foresti, G.L., Snidaro, L.: Trajectory clustering and its applications for video surveillance. In: IEEE Conference on Advanced Video and Signal Based Surveillance (2005)
23. Zhan, B., Monekosso, D.N., Remagnino, P., Velastin, S.A., Xu, L.-Q.: Crowd analysis: a survey. Mach. Vision Appl. **19**(5–6), 345–357 (2008)
24. Andrade, E.L., Blunsden, S., Fisher, R.B.: Modelling crowd scenes for event detection. In: 18th International Conference on Pattern Recognition, vol. 1, pp. 175–178 (2006)
25. Soori, U., Arshad, M.: Underwater crowd flow detection using Lagrangian dynamics. Indian J. Marine Sci. **38**, 359–364 (2009)
26. Meila, M., Shi, J.: A random walks view of spectral segmentation. In: AISTATS, pp. 8–11. AISTATS (2001)
27. Rissanen, J.: Stochastic Complexity in Statistical Inquiry, Series in Computer Science, vol. 15. World Scientific, Singapore (1989)
28. Wang, F., Jiang, Y.-G., Ngo, C.-W.: Video event detection using motion relativity and visual relatedness. In: Proceedings of ACM multimedia (2008)

29. Branson, K., Robie, A.A., Bender, J., Perona, P., Dickinson, M.H.: High-throughput ethomics in large groups of Drosophila. Nat. Methods **6**, 451–457 (2009)
30. Palmer, T., Tamte, M., Halje, P., Enqvist, O., Petersson, P.: A system for automated tracking of motor components in neurophysiological research. J. Neurosci. Methods **205**, 334–344 (2012)
31. Poppe, R.: A survey on vision-based human action recognition. Image Vision Comput. **28**, 976–990 (2010)
32. Burgos-Artizzu, X., Dollár, P., Lin, D., Anderson, D., Perona, P.: Social behavior recognition in continuous videos. In: CVPR (2012)
33. Haritaoglu, I., Harwood, D., Davis, L.S.: W4: Who? when? where? what? a real time system for detecting and tracking people. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition, vol. 1, (Nara, Japan), pp. 222–227 (2008)
34. Faro, A., Giordano, D., Spampinato, C.: Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection. IEEE Trans. Intell. Transportation Syst. **12**, 1398–1412 (2011)
35. Porikli, F.: Achieving real-time object detection and tracking under extreme conditions. J. Real-Time Image Process. **1**(1), 33–40 (2006)
36. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Cat No PR00149, **2**(c), 246–252 (1999)
37. Faro, A., Giordano, D., Spampinato, C.: Integrating location tracking, traffic monitoring and semantics in a layered its architecture. IET Intell. Transport Syst. **5**(3), 197–206 (2011)
38. Porikli, F., Wren, C.: Change detection by frequency decomposition: Wave-back. In: Proceedings of Workshop on Image Analysis for Multimedia Interactive Services (2005)
39. Porikli, F.: Multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images. In: Proceedings of IEEE Motion Multi-Workshop (2005)
40. Barnich, O., Van Droogenbroeck, M.: ViBe: a universal background subtraction algorithm for video sequences. IEEE Trans. Image Process. **20**, 1709–1724 (2011)
41. Porikli, F.: Change detection by frequency decomposition: Wave-back. In: Proceedings of Workshop on Image Analysis for Multimedia Interactive Services (2005)
42. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Mach. Learn. **37**, 297–336 (1999)
43. Kim, K., Chalidabhongse, T., Harwood, D., Davis, L.: Background modeling and subtraction by codebook construction. In: International Conference on Image Processing, 2004. ICIP '04. 2004, vol. 5, pp. 3061–3064 (2004)
44. Spampinato, C., Palazzo, S.: Enhancing object detection performance by integrating motion objectness and perceptual organization. In: Proceedings of IEEE International Conference on, Pattern Recognition, pp. 3640–3643 (2012)
45. Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. In: IEEE Transactions on PAMI, vol. 99, PrePrints (2012)
46. Cheng, C., Koschan, A., Chen, C.-H., Page, D.L., Abidi, M.A.: Outdoor scene image segmentation based on background recognition and perceptual organization. IEEE Trans. Image Process. **21**(3), 1007–1019 (2012)
47. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. ACM Trans. Graphics (TOG), pp. 309–314 (2004)
48. He, X.C., Yung, N.H.C.: Curvature scale space corner detector with adaptive threshold and dynamic region of support. In: International Conference on Pattern Recognition, vol. 2, pp. 791–794. IEEE Computer Society, Los Alamitos, CA, USA (2004)
49. Mokhtarian, F., Suomela, R.: Robust image corner detection through curvature scale space. IEEE Trans. Pattern Anal. Mach. Intell. **20**(12), 1376–1381 (1998)
50. Spampinato, C., Giordano, D., Salvo, R.D., Chen-Burger, Y.H., Fisher, R.B., Nadarajan, G.: Automatic fish classification for underwater species behavior understanding. In: Proceedings of the first ACM international workshop on analysis and retrieval of tracked events and motion in imagery streams, New York, NY, USA, pp. 45–50 (2010)
51. Chih-Chung, C., Chih-Jen, L.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 1–27 (2011)
52. Porikli, F., Tuzel, O., Meer, P.: Covariance tracking using model update based on lie algebra. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition (2005)
53. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
54. Kavasidis, I., Palazzo, S., Di Salvo, R., Giordano, D., Spampinato, C.: A semi-automatic tool for detection and tracking ground truth generation in videos. In: VIGTA '12: Proceedings of the 1st International Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications, pp. 1–5, ACM (2012)
55. Isard, M., Blake, A.: Condensation–conditional density propagation for visual tracking. Int. J. Comput. Vision **29**(1), 5–28 (1998)
56. Lazarevic-McManus, N., Renno, J., Jones, G.A.: Performance evaluation in visual surveillance using the f-measure. In: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks, VSSN '06, pp. 45–52, ACM, New York, NY, USA (2006)

Author Biographies

Concetto Spampinato received the Laurea (grade 110/110 cum laude) degree in computer engineering and the Ph.D. degree from the University of Catania, Catania, Italy, in 2004 and 2008, respectively, where he is currently Research Assistant. His research interests include mainly image and video analysis, medical image analysis and medical informatics. He has particular interest in ecological data analysis, being involved in the EU project Fish4Knowledge. He has coauthored more than 90 publications in international refereed journals and conference proceedings. As further research activities, he has organised and chaired dedicated workshops and several special sessions at mainstream conferences and guest-edited four special issues of international journals with impact factor.

Emmanuelle Beauxis-Aussalet is a PhD student at Centrum Wiskunde & Informatica in Amsterdam. Her research interests are Human–Computer Interface for data analysis, and information design for controlling the provenance of video analysis data. She received a Master in Computer Science from the Ecole Centrale Paris, and a Master in Communication through Digital Media from the Institut d'Etudes Supérieures des Arts.



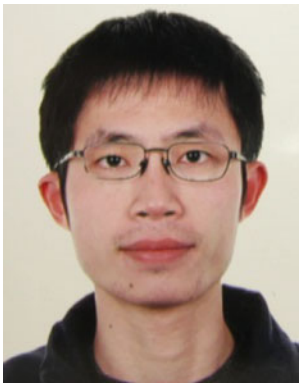
Simone Palazzo received the Laurea degree in Computer Engineering in 2010, grade 110/110 cum laude from the University of Catania, where he is currently doing his Ph.D. His interest activities include image and signal processing, image enhancement and reconstruction.

Cigdem Beyan received the BEng degree in computer engineering from Baskent University, Turkey in 2008 and MSc degree in Informatics from Middle East Technical University, Turkey in 2010. She is now a PhD candidate in School of Informatics, Institute of Perception, Action and Behaviour in University of Edinburgh, UK. She received Edinburgh Global Overseas Research Scholarship and Principal Career Development Scholarship in career area teaching. Her primary research interests are computer vision and machine learning: behaviour analysis, object detection and tracking, image sequence processing, motion analysis and pattern recognition.

Jacco van Ossenbruggen is affiliated with the Information Access group at the Centrum Wiskunde & Informatica (CWI) in Amsterdam, and with the Web & Media research group at VU University in Amsterdam. His research interests include user interfaces for unreliable data, web-based metadata modeling and integration, and data provenance on the web. He is currently researching these topics in the cultural heritage domain (as part of the European LinkedTV and the Dutch national COMMIT projects) and in the marine biology domain (in the European Fish4Knowledge project). He obtained a PhD in computer science from VU University Amsterdam in 2001.

Jiyin He is a postdoctoral researcher at the Information Access group at CWI (Centrum Wiskunde en Informatica) in Amsterdam, the Netherlands. She received her PhD degree from the University of Amsterdam. She holds M.Sc. degree in Artificial Intelligence from K.U. Leuven, Belgium. Her research includes information retrieval, query log analysis, crowd sourcing for evaluation, and user modeling for information access systems.

Bas Boom is currently a Research Associate at the University of Edinburgh in the School of Informatics. In 2005 he received the Master degree from the Free University of Amsterdam in Computer Science on a thesis entitled "Fast Object Detection". This thesis was the result of a successful internship at the company PrimeVision, where he developed methods for fast detection (localisation) of license plates, faces and addresses in images. He has received his PhD at the University of Twente in the field of face recognition with special interests in face registration and illumination correction. He has been organising several scientific workshops (VAIB 2012, VIGTA 2012 and 2013) and is the guest editor for the related special issues. He has published several journal and conference articles on biometrics and computer vision.



Xuan Huang is a PhD student at The University of Edinburgh in the School of Informatics. His academic advisors are Prof. Robert Fisher and Prof. Chris Williams. He is a member of the Institute of Perception, Action and Behaviour. His research interest is in computer vision area. He is now a member of Fish4Knowledge project and doing fish species recognition.