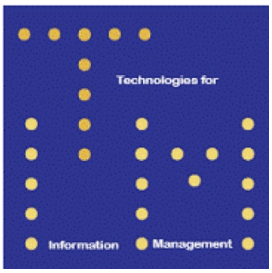
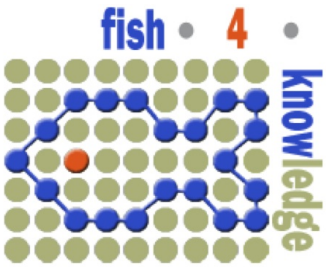




Fish4Knowledge Deliverable D1.1

Fish Detection and Tracking

Principal Author: A. Faro, D. Giordano, S. Palazzo, C. Spampinato
Contributors: UCAT
Dissemination: PU

Abstract: The Fish Detection and Tracking deliverable (D1.1) of the Fish4Knowledge project describes the algorithms devised, implemented and tested for the detection and tracking of fish in underwater videos. Several detection algorithms are needed to deal with the different issues encountered in an unconstrained underwater environment, and a covariance-based tracking algorithm has been implemented for the extraction of trajectories, which is a necessary preliminary step for accurate fish counting, event detection and behavioral analysis. Performance evaluation has been carried out by comparing the algorithms' results to hand-labeled ground truth and through self-evaluation techniques.

Deliverable due: Month 9

Contents

1 Introduction

Fish detection and tracking is a fundamental task for the whole F4K system, since it provides the basic evidence for higher-level analysis: species recognition, behaviour understanding, population statistics generation, workflow composition. However, the specific application context makes these tasks very challenging: underwater video shooting constrains the quality of the video (because of the technical difficulties in linking the cameras to mainland servers) and the targets themselves (i.e. fish) have more degrees of freedom in motion than, for example, people or vehicles in urban environments.

In order to deal with the various environmental difficulties found in underwater videos (such as light changes, murky water, swaying plants), several fish detection algorithms have been implemented: the well-known Gaussian mixture model, a mixture model variant based on Poisson distributions, an approach based on intrinsic images to deal with illumination changes, a frequency-decomposition technique to handle periodic movements of background elements, such as plants.

These algorithms are based on a background-modeling approach, which consists in estimating a "background image" (i.e. how the scene looks like without fish) and comparing each new frame to this model, in order to detect pixels which deviate from it and which are marked as foreground. Of course, since the environmental conditions are likely to change with time, model update strategies have been implemented to keep the background description up-to-date with the current scene.

The detection algorithms used in the project are described in Section 2. The reason for implementing and testing several detection algorithms is to assess the suitability of each of them to the different scene conditions in order to provide the higher processing levels (e.g. the workflow composition level) with different alternatives to use for answering user queries.

The detection performance has been improved by adding a post-processing filtering stage, in which objects are selected according to a quality score (referred in the F4K database as $detection\_certainty$) describing how sure we are that a detected blob be a fish. The computing of this score (described in Section 2.1) is based on the analysis of the color and motion vector in the proximity of an object's contour (to check whether there is a marked difference between object and background) and inside the object itself (to check for uniformity of motion and color).

The tracking algorithm relies on a covariance model (Section 3), and, similarly to the detection stage, each tracking decision comes with a quality score ($tracking\_certainty$ in the F4K database), which has been used mainly for on-line self evaluation purposes (Section 3.1). The extracted trajectories are then represented by an average quality score (average score of all tracking decisions for the specific trajectory), which is an index of the goodness of the trajectory in terms of the regularity of the motion (direction and speed smoothness) and similarity of the object's appearances (grey and color histograms and textures).

The detection and tracking quality scores have been used also to gather more data for performance evaluation without resorting to hand-labeled ground truth, which is very time-consuming to obtain.

Finally, in Section 4 we present some results on the performance of the described algorithms, and a set of statistics on the current detection and tracking data available in the database (host: $f4k-db.ing.unict.it$, database: $f4k\_db$, username: $f4k-user$, password: $f4k-pwd$) whose schema is described in Deliverable 5.2.

2 Fish detection

The characterization of video processing techniques for target detection and tracking in underwater unconstrained environment is an important task that exemplifies the topical challenges of monitoring real-life systems. In detail, detection and tracking techniques should handle different effects that usually occur in the observed scenes:

- sudden and gradual light changes: typically, the videos are available starting from sunrise until sunset, so it is necessary to consider the light transition due to these particular moments of the day when brightness and contrast of the images are strongly compromised by the absence of sunshine. Moreover, the periodical gleaming on underwater scene has to be considered when designing the detection and tracking algorithms;

- bad weather conditions: the weather conditions are subject to unexpected changes such as sudden cloudiness, storms and typhoons. Under these conditions the scene becomes very difficult to analyze due to a worsening of image contrast which makes it hard to detect and track targets accurately.

- murky water: in order to investigate the movement of fish in their natural habitat it is important to consider that the clarity and cleanness of the water flow during the day could change due to the drift and the presence of plankton. Under these conditions, targets that are not fish might be detected as false positive;

- algae on camera lens: the direct contact of seawater with lens causes a rapid formation of algae and filth that compromises the quality of the observed scene;

- periodic and multimodal background: handling background movements and variations is one of the most difficult tasks and algorithms must be robust enough to cope with arbitrary changes in the scene. Also periodic movements (e.g. plants affected by flood-tide and drift) have to be taken into account to avoid the detection of moving non-fish objects.

However, one of the most complex issues to deal with, when processing underwater videos, concerns the targets to be detected (in our case fish). Indeed, differently from humans, fish show erratic and fast movements (in three dimensions) that lead to frequently changes in size and appearance. Hence, the detection and tracking modules have to adapt the model effectively both to scene and fish appearance variations.

Although object detection is an extensively studied problem in the literature, none of the existing approaches has been demonstrated to be generally superior to the other ones, rather the performance depends on the specific application and context. Generally, a common approach for discriminating moving visual objects in a scene consists of a preliminary description of the background without any objects of interest followed by a comparison between the observed image and the background reference to obtain a foreground mask. Unfortunately in our case different approaches are needed because of the presence of periodic and multimodal background, illumination variations and arbitrary changes in the observed scene, which make difficult to build a description of the background that can tolerate such scene changes. To deal with these issues several approaches have been investigated and implemented.

Typically, to handle multi-modal backgrounds, a mixture of background models can be adopted. The background is modeled by using statistical distributions representing each pixel's

history. A pixel is considered to belong to the background if there exists any distribution with enough supporting evidence. As new frames are processed, the distributions' parameters are updated to reflect the changes which might occur in the scene. This approach is flexible enough to handle sudden and global illumination changes and other arbitrary variations in the scene. Mixture-of-model approaches can converge to any arbitrary distribution providing enough number of observations, but the computational cost grows exponentially as the number of models in the mixture increases. A drawback of these methods is that they ignore the temporal correlation of color values. This does not allow to distinguish a periodic background motion such as swaying plants driven by drift, algae on camera lens, rotating objects and so on, from the foreground object motion.

We implemented two algorithms based on this mixture-of-model approach, the well-known "Adaptive Gaussian Mixture Model" (AGMM) [1] and the "Adaptive Poisson Mixture Model" (APMM) [2]. AGMM, which uses Gaussian distributions to model the background, deals very well with videos containing multi-modal backgrounds but it cannot handle frequent or abrupt lighting changes. APMM uses Poisson distributions instead, based on the observation that the intensity of pixels is Poisson-distributed due to the natural variation of photons density hitting the sensors.

Since real-world physics often induces near-periodic phenomena in the environment, frequency decomposition-based representations of the background have been proposed [3] . These algorithms detect objects by comparing the frequency transform representations of the background and the current scene. This requires to keep a fairly large number of past frames in order to accurately model the current frequency map against which to compare the background model.

The "Wave-back" algorithm [3] uses frequency decomposition of each pixel's history to catch periodic background movements (typical of underwater plants). In particular, given the set of previous frames, the DCT (Discrete Cosine Transform) coefficients of each pixel are calculated and are compared to the respective background coefficients resulting in a distance map. By thresholding the distance map, the foreground objects can be identified. The "Wave-back" algorithm performs well in videos with repetitive scenes and low-contrast colors but cannot deal adequately with videos with erratic and fast fish movement and when sudden lighting transitions take place.

A technique devised to handle problems related to noise and illumination changes consists of representing the scene using intrinsic images, where each image is obtained as a multiplication between its static and dynamic components [4]. This approach has been used as a basis for another algorithm we implemented, called "Intrinsic Model", which performs the multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images. Every image is split into two components: the reflectance image (static component) and the illumination image (dynamic component). The former is invariant to lighting changes and will be almost identical under any light conditions. The background is modeled by calculating the temporal median of these components. This algorithm performs well in videos with lighting changes, fast and erratic fish movements or low-contrast but it suffered when multi-modal backgrounds (e.g. algae) were present.

2.1 Detection Quality Score

Because of the problems associated in handling false positives and false negatives, we introduced a post-processing stage, which assigns a quality score to each object detected during execution. This score is a numerical value computed by estimating the likelihood that a detected object is actually a fish using *a-priori* knowledge on the shape, color and boundary features of the object. This score, besides providing feedback data according to which it is possible to tune the algorithms in order to improve their accuracy, has also been used to filter out objects with low scores (see Section 4 for performance comparisons).

The overall score is a value included between 0 and 1 (the larger, the more certain), computed as the average of the following features:

- *Difference of color at object boundary*: this index is based on the assumption that color boundaries often coincide with fish/background boundaries; this value is highest when the areas "just inside" and "just outside" of the contour of the detected object have markedly different color values. It is computed according to the following procedure.
  - Select N (currently, 25) equidistant points on the object's contour.
  - For each point P_i,
    - Select two points $P_{i,in}$ and $P_{i,out}$ located just inside and just outside of the contour on the line passing by P_i and such that it is orthogonal to the tangent of the contour in that point.
    - Compute $C_{i,in}$ and $C_{i,out}$ as the average color intensities in the 5×5 neighborhoods of $P_{i,in}$ and $P_{i,out}$.
  - Compute the result as:

$$\Delta_{CO} = \frac{1}{N} \sum_{i=1}^{N} \frac{||C_{i,in} - C_{i,out}||}{\sqrt{3 \cdot d^2}} \tag{1}$$

    where the denominator represents the minimum Euclidean distance between two pixels to be considered as belonging to markedly separate color regions (d is the corresponding distance for a single color channel; its current value is 75).

- *Difference of motion vectors at object boundary*: similarly to the previous case, the motion vector in the regions close to the object contour are compared; this value takes into account the fact that, hypothetically, the motion vector outside of the object contour should be zero (static background) or significantly different than inside (as shown in Fig. 1). In the following formula, $M_{i,in}$ and $M_{i,out}$ represent the average motion vectors computed just inside and just outside of contour point P_i (obtained as in the previous formula).

$$\Delta_{MV} = \frac{1}{N} \sum_{i=1}^{N} \frac{||M_{i,in} - M_{i,out}||}{||M_{i,in}|| + ||M_{i,out}||} \tag{2}$$

- *Internal color homogeneity*: due to the low resolution of videos, most fish (especially the ones far away from the camera) appear as monochromatic, so this index gives an indication on how homogenous the color distribution of the detected fish is. The body

of the object is divided into a grid and for each cell the average color value is computed; the more similar these average results are, the more likely it is that the detected object is actually a fish. This value is computed as follows:

  - For each cell j in the grid, compute the mean color C_j.
  - Compute C_M as the mean of all $\{C_j\}$.
  - For each C_j, compute $d_j = ||C_j - C_M||$.
  - Compute d_M as the mean of all $\{d_j\}$.
  - Return $\left(-\alpha_{ICH} d_M + \beta_{ICH}\right) / \gamma_{ICH}$.

- *Internal motion homogeneity*. This index is based on the assumption that the internal motion vectors of a correctly-detected fish are more uniform than the ones of an erroneous fish detection (e.g. a moving plant), as shown in Fig. 2. Similarly to the previous case, the object is divided into a grid and the average motion vectors for each cell are compared. However, the computation of the corresponding score is slightly different:

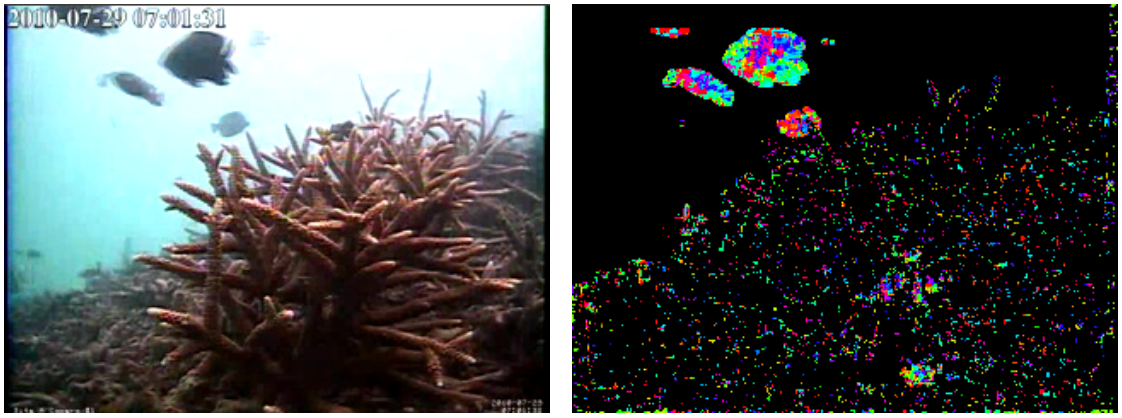
  - Given an object, its current bounding box and the bounding box of its last appearance, compute the motion vector of the region R obtained as the union of the two bounding boxes.
  - For each motion vector point in R, mark it as "not valid" if its displacement projects the point out of R. This might happen because the motion vector algorithm is not accurate for points belonging to the common region between two detections.
  - For each cell j in the grid which has at least one valid motion vector point, compute the mean motion vector V_j, whose components are the average Δx and Δy.
  - Compute the variances v_x and v_y of the first and second components of all $\{V_j\}$.
  - Return $1 - \frac{v_x + v_y}{\alpha_{IMH}}$.

- *Internal texture homogeneity*: again, an internal grid is defined and a set of Gabor filters is applied to each cell; this score aims at checking that the object presents a uniform texture.

All symbols such as α_{ICH} in the descriptions above represent constant values which are experimentally identified to normalize the results between 0 and 1.

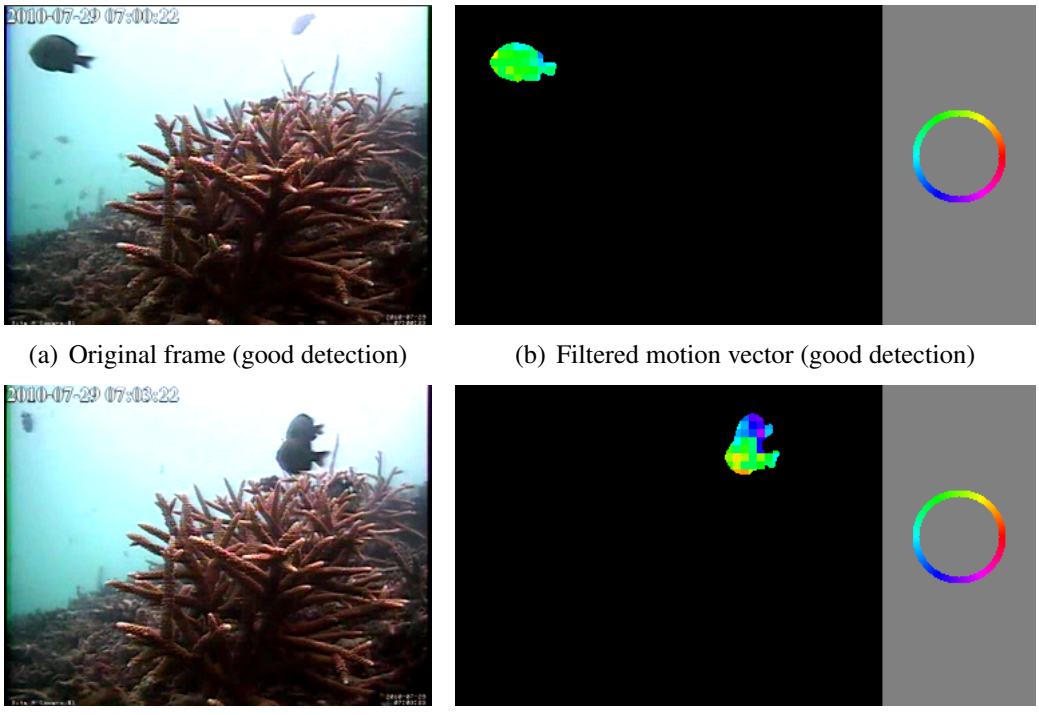
To give a visual idea of the how such scores get mapped to actual detections, Figure 3 shows a few examples of detection with the associated quality score.

The detection score as well as the tracking score has been also adopted for on-line performance evaluation without resorting to hand-labeled ground truth. After the fish detection stage, tracking is carried out. Tracking serves both as a proof of the correctness of the detected fish – i.e. a moving fish usually remains in the monitored scene longer than a spurious object and its movements (though erratic) are more regular than plants' ones – and to consistently extract fish paths, which are the basis for the subsequent event detection and behaviour understanding modules.



(a) Original frame (b) Motion vector

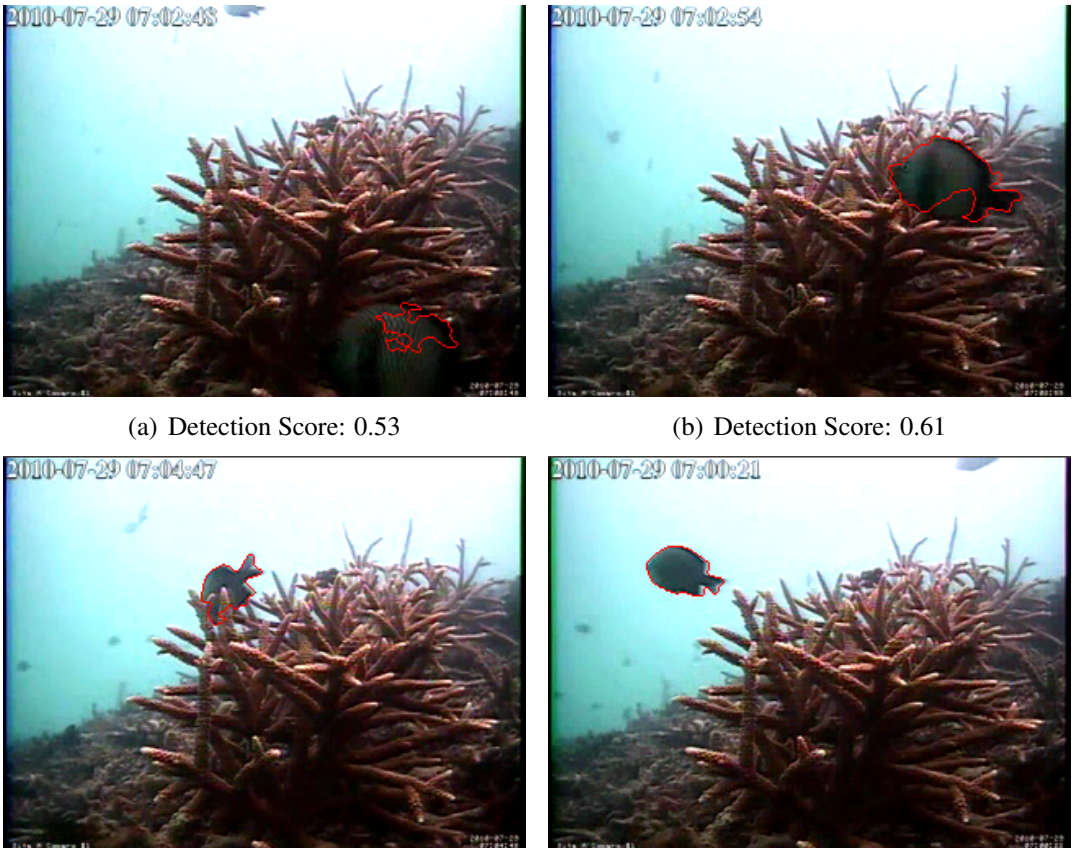
Figure 1: Motion vector boundaries typically coincide with object boundaries.



(a) Original frame (good detection) (b) Filtered motion vector (good detection)

(c) Original frame (bad detection) (d) Filtered motion vector (bad detection)

Figure 2: Correct object detections are made up of points generally orientated towards the same direction. The circle to the right of subfigures (b) and (d) maps color to direction of motion.



(a) Detection Score: 0.53 (b) Detection Score: 0.61

(c) Detection Score: 0.75 (d) Detection Score: 0.89

Figure 3: Examples of detection quality scores.

3 Fish tracking

Visual tracking consists of following an object in a video across consecutive frames; in other words, a tracking algorithm has to be able to recognize that two regions in two different frames represent the same object.

However, this task presents a few major difficulties, which are even more difficult to manage in unconstrained environments such as the one we are dealing with in the F4K project. First of all, the search region of an object has to be limited to a neighborhood of its previous detection, otherwise a new object appearing in the scene might be associated to the old one, even if it is located in a different area of the video. The choice of the search area depends on the motion characteristics of the objects which typically appear in the scene. For example, in a urban environment we can safely assume that pedestrians and cars move approximately always in the same direction, without swift changes, and this allows to restrict the search area to something shaped like a cone oriented towards the main direction of the target. However, with fish this is not necessarily true, because their typical erratic motion makes their direction less predictable, especially in videos with low frame rate (for example, because of the difficulty in the transmission of large amounts of data from underwater cameras to a central storage server). Another tracking issue consists of the change in appearance of an object across the video, because of variations of lighting, orientation, shape. This is especially true for fish, because of their non-rigidity and, again, of erratic motion. Finally, another complication is caused by occlusions, that is partial or total overlap of two objects, such that one of them is hidden by the other one.

The approach adopted in this work for designing the tracking algorithm consists of:

- Identification of objects of interest (performed by the above-described detection algorithms).
- Extraction of a feature set from each identified object, used as a model to compare currently tracked objects with the new detections in the scene.
- Association between tracked objects and detections in the scene, according to a best-matching policy.

Many approaches have been studied in the literature on how to solve the visual tracking problem [5]. Among these, the most famous and widely-used algorithms are Kalman filter-based tracking [6], particle filter tracking [7], point feature tracking [8], mean shift tracking [9].

Tracking algorithms based on covariance representation [10] model objects as the covariance matrices of a set of feature built out of each pixel belonging to the object's region. This representation takes into account both the spatial and statistical properties, unlike histogram representations (which disregard the structural arrangement of pixels) and appearance models (which ignore statistical properties). In the context of F4K, fish tracking is necessary to consistently count the number of unique fish in the video (which of course is less than the number of total appearances) and to provide data for the following stage of event detection and behavior understanding based on trajectory analysis.

The tracking algorithm chosen to handle all the phenomena typical of underwater domain, is based on [11] and uses covariance matrices computed on a set of pixel-based features to model the fish appearance.

In the following description, we use "tracked object" to indicate an entity that represents a unique fish and contains information about the fish appearance history and its current covariance model; and "detected object" to indicate a moving object, which has not been associated to any tracked object yet. For each detected object, the corresponding covariance matrix is computed by first building a feature vector for each pixel, made up of the pixel coordinates, the RGB and hue values and the mean and standard deviation of the histogram of a 5×5 window which contains the target pixel. The covariance matrix, which models the object, is then computed from this feature vector and associated to the detected object. Afterwards, the covariance matrix is used to compare the object with the currently tracked objects, in order to decide which one it resembles the most. For each tracked object, its search area (i.e. the region of the image inside which the algorithm looks for detected objects which can be associated to that tracked object) is made up of two rectangles: a "central search area", which is a rectangle centered at the object's previous location and that expands evenly in all directions; and a "directional search area" which takes into consideration the recent motion trend of the object to give a simple estimate of its direction, and which consists of a rectangle with a vertex at the object's previous location and the correspondent opposite vertex located according to the estimated object's direction.

The main issue in comparing covariance matrices is that they do not lie on the Euclidean space—for example, the covariance space is not closed under multiplication with negative scales. For this reason, we use Förstner's distance [12], which is based on generalized eigenvalues, to compute the similarity between two covariance matrices

The model of each tracked object is then computed as a mean of the covariance matrices corresponding to the most recent detections of that object. In order to deal with occlusions, the algorithm handles the temporary loss of tracked objects, by keeping for each of them a counter of how many frames it has been missing; when this counter reaches a user-defined value, the object is considered lost and discarded. Of course, it is important to find a good trade-off for this value: if it is too low, an object which temporarily disappears (for example, because it is hidden behind a plant) might be treated as a different object when it reappears; if it is too high, different objects might be recognized as the same one, which has long exited the scene. Moreover, this value also influences the size of the search area: the longer an object has been missing, the wider its search area will be. The current value of this parameter, optimized for 5-fps videos, is 6.

The results obtained with this algorithm show that it can accurately follow a fish even when it is temporarily hidden behind plants or in presence of similar fish in the scene. However, the accuracy of the algorithm is strongly linked to that of the detection algorithm, since it assumes that all and only moving objects will be provided by the underlying motion algorithm; for this reason, tracking may fail because of detection inaccuracy.

3.1 Tracking Quality Score

The fish tracking algorithm is one of the most important components of the proposed system since it identifies the fish trajectories on which the behavior understanding and event detection modules will rely. Thereafter, in order to consistently investigate fish behavior, it is necessary to estimate the quality of each detected trajectory and to select the ones that respect a specific criteria of goodness, thus avoiding to invalidate the final behavior analysis. To compute such a quality score (referred in the following as q_S), we have adopted a series of measurements taken from [13] and [14] and combined them with new measurements in order to obtain values

indicating the goodness and the plausibility of a trajectory. In detail, for each tracking decision (i.e. an association between an object in frame t and one in frame $t+1$) the following features are computed and combined into a single score as an average.

- *Difference of shape ratio between frames*: this score detects rapid changes in the object's shape, which might indicate tracking failure. This value is high if the shape ratio ($R = \frac{W}{H}$, with W and H, respectively, the width and the height of the bounding box containing the object) between consecutive frames $t-1$ and t keeps as constant as possible.

$$\begin{aligned} R_{max} &= max\left\{R_t, R_{t-1}\right\} \\ R_{min} &= min\left\{R_t, R_{t-1}\right\} \\ shape\_ratio\_score &= \alpha_{SR}\frac{R_{min}}{R_{max}} \end{aligned}$$

- *Histogram difference*: this feature evaluates the difference between two appearances of the same object by comparing the respective histograms (analyzing independently the three RGB channels and the grayscale versions of the two objects). Given histograms H_t and H_{t-1}, the corresponding score is proportional to the ratio between the intersection and the union of the two histograms:

$$\alpha_{HD}\sum_{i=0}^{255}\frac{min\left\{H_t\left(i\right), H_{t-1}\left(i\right)\right\}}{max\left\{H_t\left(i\right), H_{t-1}\left(i\right)\right\}}$$

- *Direction smoothness*: although fish movements are erratic, we can safely assume that a fish trajectory is as good as it is regular and without sudden direction changes in the short term (i.e. in few consecutive frames). This value keeps track of the direction of the object in the last frames and checks for unlikely changes in the trajectory. It is computed as:

$$direction\_smoothness = 1 - \frac{|\theta_1 - \theta_2|}{180}$$

  where θ_1 and θ_2 are the angles (with respect to the x axis) of the last two displacements of the object. For simplicity, we use $\theta_1 - \theta_2$ in the formula, although the actual implementation handles the case of angles around the $0°/360°$ boundary.

- *Speed smoothness*: similarly to the previous feature, this value checks whether the current speed of the object (i.e. the displacement between the previous position and current one) is similar to the average speed in the object's history. Let P_t and P_{t-1} be the last two positions of the object, we compute $s_t = ||P_t - P_{t-1}||$, so that s_t represents the last displacement (speed) of the object, and compare it with the average speed $\bar{s}$ in order to compute $speed\_smoothness$ as:

$$\begin{aligned} s_{max} &= max\left\{s_t, \bar{s}\right\} \\ s_{min} &= min\left\{s_t, \bar{s}\right\} \\ speed\_smoothness &= \alpha_{SS}\frac{s_{min}}{s_{max}} \end{aligned}$$

- *Texture difference*: texture features (mean and variance of several Gabor filters) are computed from two consecutive appearances and compared. Given two feature vectors v_1 and v_2, this value is computed as:

$$1 - \sqrt{\sum_{i=1}^{n} \left(v_1\left(i\right) - v_2\left(i\right)\right)^2} / \alpha_{TD} \tag{3}$$

- *Temporal persistence*: this value is the number of frames in which a given object appears.

Similarly to the detection evaluation scores, all constants included in the above formulas are experimentally identified to normalize the resulting values between 0 and 1. The overall quality score q_S is computed as follows:

- Compute the average μ of the above-described values, except the temporal persistence TP.
- If $TP > N$, return μ;
- Else, return $\mu \cdot \left(0.9 + \frac{TP}{10 \times N}\right)$ (i.e. if an object appears in less than N frames, limit the maximum score it can get).

Fig. 4 shows a few sample trajectories with related average q_S scores; it is possible to notice that the trajectory of the top-left image is unrealistic (and is caused by a temporary mis-association between objects) and its average score, computed as average of the scores of each tracking decision for all the appearances of a fish (four times, in this case), was 0.63, whereas the image on the top-right side shows a correct trajectory whose average score is of 0.91. The two bottom images show, from left to right, a complex but correct trajectory (with a 0.81 score) and a trajectory which is correct up to a certain point, before an occlusion happened, so its total tracking score is 0.71.

4 Experimental results

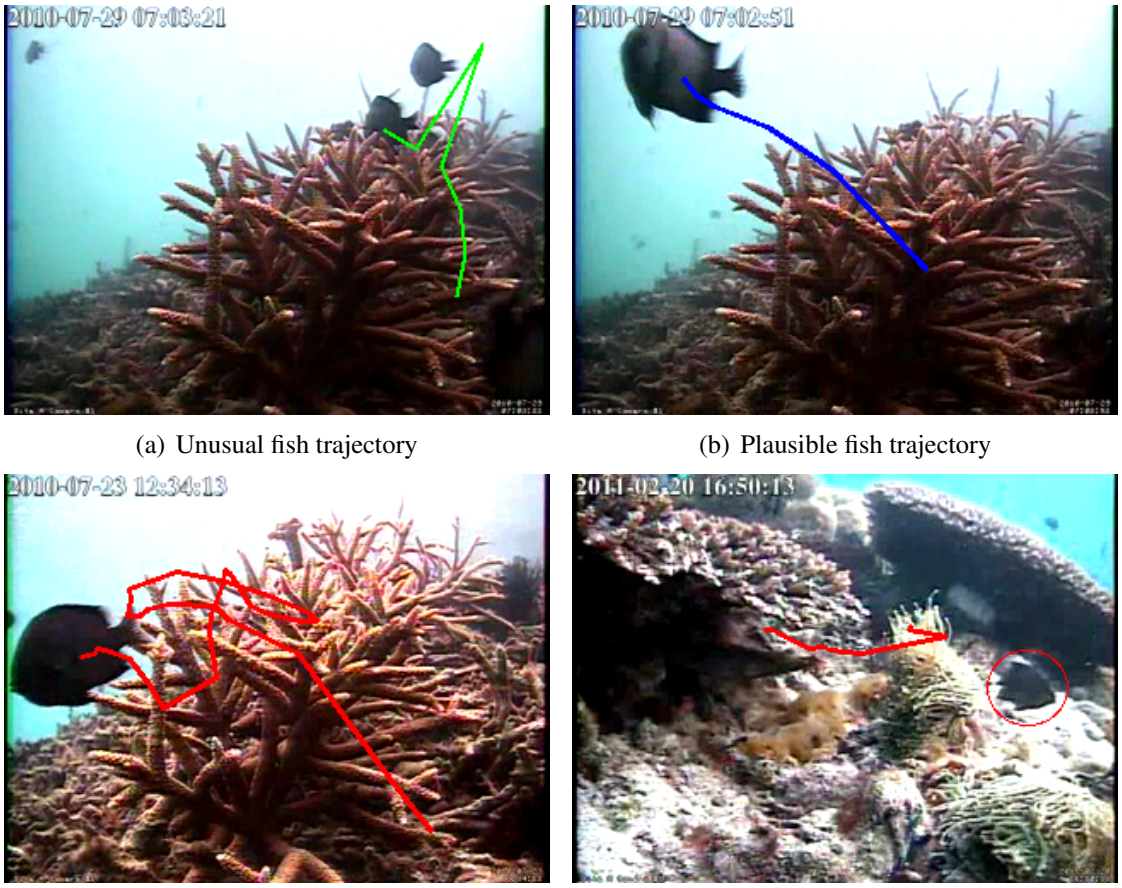
In order to evaluate the performance of the four current algorithms for fish detection, i.e. Adaptive Gaussian Mixture Model (AGMM), Adaptive Poisson Mixture Model (APMM), Intrinsic Model (IM) and Wave-back (WB), and of the covariance-based tracker, together with a post-processing module for object filtering, we have compared the results obtained by the algorithms to a set of manually-labeled ground-truth videos.

To evaluate the performance of the detection algorithms we adopted common metrics, i.e. detection rate (DR) and false alarm rate (FAR) which are defined as:

$$DR = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{4}$$

$$FAR = \frac{N_{FP}}{N_{TP} + N_{FP}} \tag{5}$$

where N_{TP}, N_{FP} and N_{FN} are, respectively, the number of true positives, false positives and false negatives.



(a) Unusual fish trajectory (b) Plausible fish trajectory

(c) Plausible fish trajectory (d) Fish-plant Occlusion

Figure 4: Examples of fish trajectories: (a) an erroneous path (average q_S score 0.63) due to a failure of the tracking algorithm; (b) a correct path (average q_S score is of 0.91); (c) a complex but correct fish path with an average q_S score of 0.81; (d) trajectory with an average q_S score of 0.71 due to a fish-plant occlusion.

Video	Description	N_F
1	Dynamic Background Striped Fish Texture	156
2	Dynamic Background Camouflage phenomena	1373
3	Typhoon Frequent illumination variations Very low contrast	1790
4	Typhoon Plants movements	34
5	High illumination Camouflage phenomena Striped Fish Texture	840

Table 1: Description of the videos used as ground truth

For the evaluation of the detection and tracking performance we used five videos (10 minutes long each) of the Fish4Knowledge repository. The videos had resolutions of 320×240 with a 24-bit color depth at a frame rate of 5 fps. The selection of these videos was based on the presence of specific features to test the effectiveness of every algorithm when non-standard conditions were encountered. In particular, the features taken into account were: dynamic backgrounds, illumination variations, high water turbidity, low contrast and camouflage phenomena. The ground truth on these videos was hand labeled. The videos are described in Table 1 together with the number of hand-labelled fish (N_F) in the ground truth data.
The performance evaluation of the detection algorithms was carried out both *at blob level* to test their ability in detecting effectively objects and *at pixel level* to test their capabilities in preserving objects' shape.
The performance, at blob level, was evaluated by testing the detection algorithms when run with and without the post-processing module (i.e. by using or not quality scores). When this module was disabled, the blobs detected by the object detection algorithms were filtered according to simple shape and area criteria, in order to exclude objects which were too small or whose shape made it unlikely to be a fish. When the post-processing module was used, no shape or area filtering was performed, however objects with a detection quality score lower than 0.65 were discarded. To give an idea of the accuracy, Table 2 shows the achieved results with and without the post-processing module at the best operating point. It is possible to see a sensible improvement in both the detection rate (by about 15%, in average) and the false alarm rate (reduced by about 10%) when using the post-processing module, showing how simple discrimination criteria based on the analysis of the size and shape of an object are not enough for a good blob filtering.

To test the performance at pixel level, we assessed the number of pixel-wise true positives (pixels correctly detected as belonging to the fish), false positives (background pixels detected as part of the object) and false negatives (object pixels mistaken for background) for each object correctly identified by a detection algorithm. According to these values, we then computed the pixel detection rate (PDR) and the pixel false alarm rate (PFAR) and the average values for each ground-truth video are shown in Table 3.

We can see how all the algorithms have a good pixel detection rate, i.e. they are able to correctly identify pixels belonging to a fish, however the results also show a relatively high pixel false alarm rate (background pixels included into the objects' blobs), especially IM and

		Without Post-processing		With Post-processing	
		DR	FAR	DR	FAR
Video 1	AGMM	65%	27%	83%	8%
	APMM	72%	21%	85%	10%
	IM	71%	11%	85%	7%
	WB	52%	21%	69%	5%
Video 2	AGMM	70%	9%	82%	7%
	APMM	67%	15%	85%	8%
	IM	73%	11%	89%	6%
	WB	75%	6%	70%	5%
Video 3	AGMM	70%	15%	81%	10%
	APMM	61%	12%	84%	6%
	IM	67%	9%	86%	9%
	WB	55%	18%	71%	8%
Video 4	AGMM	76%	28%	88%	17%
	APMM	65%	33%	85%	9%
	IM	68%	26%	91%	11%
	WB	47%	36%	63%	18%
Video 5	AGMM	67%	10%	82%	9%
	APMM	71%	17%	82%	6%
	IM	72%	22%	89%	7%
	WB	60%	19%	71%	4%

Table 2: Experimental results achieved when the detection algorithms run with and without the post-processing module

WB and mainly when the contrast of the video is low.

In order to compare the tracker' performance to the actual results, tracking identification numbers (IDs) were added to the ground-truth information to label detections in different frames as belonging to the same unique fish, in such a way that fish which underwent temporary occlusions would then be re-assigned to the same tracking ID. We directly fed our tracking algorithm with the detection data (i.e. the fish) provided by the ground truth, so that the tracking results would not be influenced by detection errors. The performance of our algorithm was also compared with the one achieved by the classic CAMSHIFT algorithm (chosen because it is the only approach tested on underwater videoclips in [15]). To assess the *ground-truth-vs-algorithm* comparison we adopted the following metrics, which are based on the ones existing in the literature [16], but that describe the performance of a tracking algorithm both globally, at the trajectory level (e.g. the correct counting rate and the average trajectory matching), and locally, at the single tracking decision level (e.g. the Correct decision rate):

- *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 Figure 5).

Video	Algorithm	PDR	$PFAR$
1	AGMM	90.7%	12.5%
	APMM	88.7%	12.9%
	IM	84.4%	13.8%
	WB	94.9%	18.1%
2	AGMM	94.0%	17.7%
	APMM	97.2%	25.7%
	IM	89.4%	26.6%
	WB	98.5%	29.8%
3	AGMM	93.5%	26.9%
	APMM	89.5%	22.1%
	IM	89.3%	29.3%
	WB	96.1%	31.3%
4	AGMM	92.9%	20.4%
	APMM	92.4%	19.9%
	IM	85.4%	39.0%
	WB	89.9%	35.6%
5	AGMM	91.7%	13.0%
	APMM	95.8%	26.3%
	IM	88.8%	17.0%
	WB	93.5%	26.3%

Table 3: Segmentation results obtained by the detection algorithms on the ground-truth videos

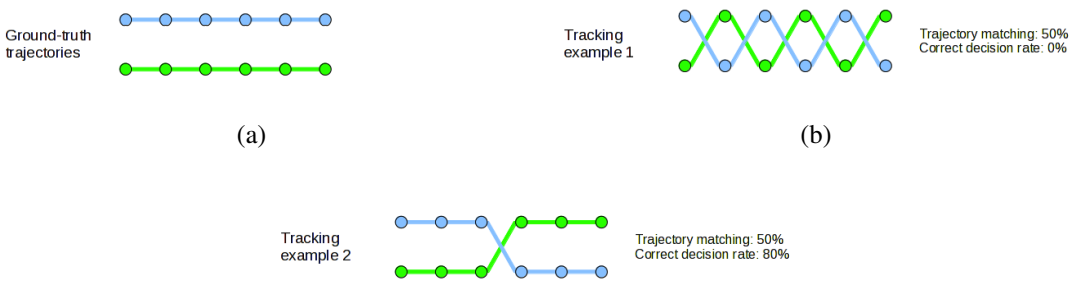


Figure 5: Difference between the trajectory matching score and the correct decision rate. Fig. 5(a) shows two ground truth trajectories of two fish, whereas the other two images represent two examples of tracking output. In Fig. 5(b), although the tracker fails at each tracking decision, the trajectory matching score is relatively high (50%), whereas the correct decision rate is 0. Differently, in Fig. 5(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).

Table 4 shows the results obtained by the covariance tracking algorithm compared to the ones achieved by the CAMSHIFT algorithm, in terms of the above-described indicators. It is clear how our approach performs better than CAMSHIFT and also has a very good 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.

	Covariance tracker	CAMSHIFT
CCR	91.3%	83.0%
ATM	95.0%	88.2%
CDR	96.7%	91.7%

Table 4: Comparison between the results obtained by the proposed algorithm and CAMSHIFT on the ground-truth data.

5 Database Content Overview

Further results have been obtained by analyzing the quality scores computed by the system and the distribution of the detection entries in the database, among the different camera locations. These results are shown in tables from 5 to 9 and refer to the database's content as available on October 15th, 2011. Most of the processed videos are taken from the NPP-3 location and very few from Lan-Yu. This is mainly because there is a higher availability of 320×240-resolution videos in NPP-3 and we are not currently processing 640×480 videos because of the low quality (however, at the time of writing, work is in progress to assure the availability of high-frame-rate and high-quality 640×480 videos). Let us recall that each processed video is 10 minutes long, with a 24-bit color depth at a frame rate of 5 fps. The videos were recorded by using 8 underwater cameras organised per location, namely NPP-3 (four cameras), HoBiHu (three cameras) and Lanyu (one camera).

Table 5 depicts the total number of processed videos, detections and fish of the four detection algorithms, whereas Tables 6 and 7 show the categorisation, respectively, by algorithm and location.

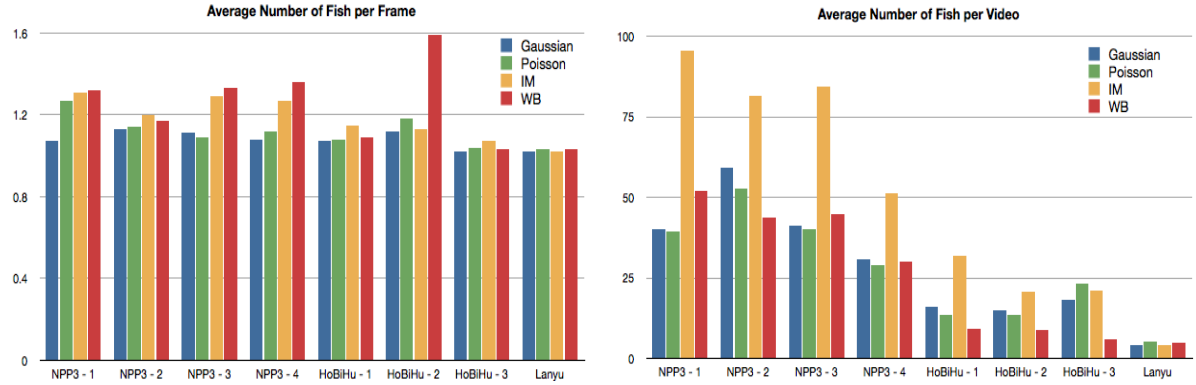
Number of processed videos	2825
Number of detections	3869473
Number of fish	456622

Table 5: Total number of processed videos, detections and fish for all the detection algorithms and all the locations in the F4K database

	AGMM	*APMM*	*IM*	*WB*
Number of processed videos	2825	2825	2825	2825
Number of detections	731049	708292	1326058	1104074
Number of fish	97267	91925	177609	89821

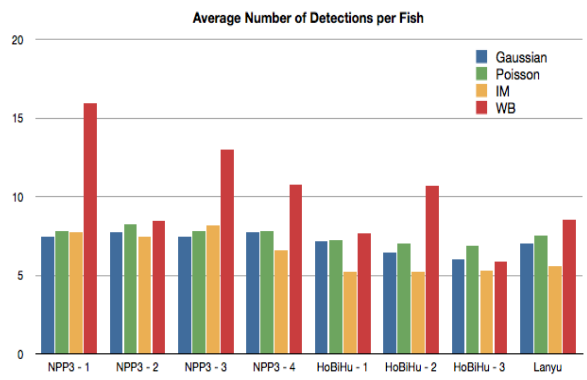
Table 6: Number of processed videos, detections and fish by algorithm

Fig. 6, instead, shows the distribution by camera and algorithm of the average number of fish per frame (Fig. 6(a)) and per video (Fig. 6(b)) and of the average number of detections per video (Fig. 6(d)) and per fish (Fig. 6(c)).

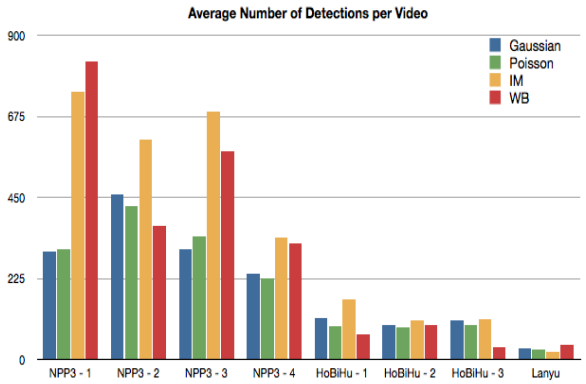


(a)

(b)



(c)



(d)

Figure 6: Average results by camera and algorithm.

	NPP-3	*HoBiHu*	*Lanyu*
Number of processed videos	2367	545	138
Number of detections	1007794	43926	3572
Number of fish	123528	7753	603

Table 7: Number of processed videos, detections and fish by location

As aforementioned ground-truth generation cannot be performed on a large number of videos, as it is a very time-consuming and error-prone operation; for this reason we also performed an on-line performance evaluation of detection and tracking algorithm on the processed videos (unlabeled) by computing the mean and the standard deviation of the quality scores assigned to each detection and trajectory extracted by the system. A comprehensive evaluation of the reliability of the quality scores is beyond the aim of this document, however, we compared the detection and tracking scores achieved by our algorithms with the ones obtained on high quality hand-labeled ground truth data and the results are shown, respectively, in Tables 8 and 9. Let us note that we assessed the tracking quality scores by taking into account the detection algorithms (namely, AGMM, APMM, IM and WB) together with our covariance-based tracking approach, since tracking quality score, unlike the detection one (which relies only on detection algorithms), is influenced by both detection and tracking.

	NPP-3				HobiHu			Lanyu
	1	2	3	4	1	2	3	1
AGMM	0.80± 0.10	0.82± 0.09	0.79± 0.09	0.74± 0.08	0.74± 0.08	0.70± 0.07	0.70± 0.07	0.73± 0.09
APMM	0.76± 0.09	0.82± 0.11	0.76± 0.09	0.68± 0.05	0.79± 0.08	0.70± 0.07	0.70± 0.06	0.68± 0.06
IM	0.74± 0.10	0.79± 0.10	0.74± 0.08	0.71± 0.07	0.66± 0.08	0.67± 0.07	0.68± 0.07	0.74± 0.09
WB	0.67± 0.08	0.68± 0.10	0.66± 0.08	0.68± 0.06	0.73± 0.11	0.69± 0.07	0.70± 0.07	0.64± 0.07
Ground Truth Object	0.75± 0.09	0.78± 0.07	0.80± 0.08	0.74± 0.05	0.75± 0.08	0.77± 0.08	0.66± 0.04	0.70± 0.05

Table 8: Average and standard deviation of the detection quality scores achieved by the detection algorithms categorised by camera and algorithm.

	NPP-3				HobiHu			Lanyu
	1	2	3	4	1	2	3	1
AGMM	0.80± 0.11	0.81± 0.11	0.80± 0.11	0.79± 0.12	0.81± 0.11	0.79± 0.13	0.79± 0.13	0.84± 0.10
APMM	0.75± 0.14	0.79± 0.13	0.79± 0.13	0.76± 0.15	0.82± 0.12	0.79± 0.13	0.72± 0.17	0.77± 0.13
IM	0.77± 0.12	0.79± 0.12	0.76± 0.12	0.77± 0.13	0.78± 0.14	0.78± 0.14	0.78± 0.13	0.85± 0.10
WB	0.74± 0.11	0.79± 0.11	0.74± 0.11	0.75± 0.12	0.80± 0.12	0.81± 0.12	0.80± 0.12	0.83± 0.11
Ground Truth Trajectory	0.77± 0.11	0.78± 0.10	0.74± 0.12	0.77± 0.12	0.78± 0.12	0.75± 0.13	0.77± 0.13	0.81± 0.11

Table 9: Average and standard deviation of the tracking quality scores achieved by the detection algorithms categorised by camera and algorithm.

6 Conclusions

In this deliverable we have described the work done for fish detection and tracking. The results of both stages are satisfactory, especially considering the difficulties of underwater

environments. For detection we have developed four different approaches that deal with the presence of periodic and multimodal background, illumination variations and arbitrary changes in the observed scene. Although such methods have demonstrated good performance they are not able to handle the cases when all the above phenomena appear at the same time. None of the developed approaches seems to be generally superior to the other ones, even though the approaches that rely on modeling background pixels with a pdf (AGMM and APMM) represent the best compromise for all the considered scenarios. We are now investigating an approach that does not opt for a particular form of the probability density function, as deviations from the assumed model are ubiquitous. The idea is that each background pixel is modeled with a set of samples of values rather than with an explicit pixel model. A pixel is then classified as background if the number of samples from which the color distance is low is higher than a certain threshold. However, we also noticed that the detection performance is influenced by the low video resolution of initial set of videos, therefore, we aim to achieve better detection results once videos with higher resolution (both spatial and temporal) will be available. Moreover, we are also working on the integration of the detection algorithms by using AdaBoost integrated with local histogram/texture features [17] in order to find the optimal decision (optimal ROC curve) rule for the detection step.

Unlike detection algorithms, the covariance based tracker is very reliable achieving an average performance (about 95%), estimated taking into account scores computed at the trajectory level and at the at the single tracking decision level. Currently, the trajectories of fish are described using a simple point-based representation [18], i.e. a fish path is a sequence of 2D coordinates. Since this representation does not describe the real fish movement and may affect higher level video analysis such as event detection and behaviour understanding, we are working on a more comprehensive 3D spatio-temporal representation of trajectories that comprises speed, direction, contour points and fish size.

Finally, we are also investigating techniques for background objects extraction which rely on image segmentation and description [19] in order to establish possible correlations between fish, its trajectories and the static scene (e.g. corals, plants, etc.). Furthermore, since the behavior of a fish is often correlated to the behavior of another fish (e.g. predator fish chasing small fish), future work will be directed on correlating temporal and spatial events among different fish species.

References

[1] C Stauffer and W E L Grimson. Adaptive background mixture models for real-time tracking. *Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Cat No PR00149*, 2(c):246–252, 1999.

[2] A. Faro, D. Giordano, and C Spampinato. Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection. *Appearing on IEEE Transactions on Intelligent Transportation Systems*, 2011.

[3] F. Porikli and C. Wren. Change detection by frequency decomposition: Wave-back. In *Proc. of Workshop on Image Analysis for Multimedia Interactive Services*, 2005.

[4] Fatih Porikli. Multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images. In *in Proc. of IEEE Motion Multi-Workshop*, 2005.

[5] Z.H. Khan and I.Y.-H. Gu. Joint feature correspondences and appearance similarity for robust visual object tracking. *IEEE Transactions on Information Forensics and Security*, 5(3):591–606, 2010.

[6] Arnaud Doucet, Nando De Freitas, and Neil Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[7] N.J. Gordon, A. Doucet, and N.D. Freitas. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control*, 24(6):843–854, 1979.

[8] J. Shi and C. Tomasi. Good features to track. *Proc. IEEE Int. Conf. Comp. Vision and Pattern Recognition*, pages 593–600, 2008.

[9] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[10] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. *Proc. 9th European Conf. on Computer Vision*, 2006.

[11] Fatih Porikli, Oncel Tuzel, and Peter Meer. Covariance tracking using model update based on lie algebra. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[12] W. Forstner and B. Moonen. A metric for covariance matrices. Technical report, Dept. of Geodesy and Geoinformatics, Stuttgart University, 1999.

[13] Duc Phu Chau, François Bremond, and Monique Thonnat. Online evaluation of tracking algorithm performance. In *The 3rd International Conference on Imaging for Crime Detection and Prevention*, December 2009.

[14] C.E. Erdem, A. Murat Tekalp, and B. Sankur. Metrics for performance evaluation of video object segmentation and tracking without ground truth. *Proceedings of Internation Conference on Image Processing*, 2:69–72, 2001.

[15] Concetto Spampinato, Yun-Heh Chen-Burger, Gayathri Nadarajan, and Robert B. Fisher. Detecting, tracking and counting fish in low quality unconstrained underwater videos. In *VISAPP (2)*, pages 514–519, 2008.

[16] F. Bashir and F. Porikli. Performance Evaluation of Object Detection and Tracking Systems. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS2006)*, June 2006.

[17] Ivan and Laptev. Improving object detection with boosted histograms. *Image and Vision Computing*, 27(5):535 – 544, 2009.

[18] J. Owens and A. Hunter. Application of the self-organising map to trajectory classification. In *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, pages 77 –83, 2000.

[19] A. Bosch, X. Muoz, and J. Freixenet. Segmentation and description of natural outdoor scenes. *Image and Vision Computing*, 25(5):727 – 740, 2007.

