Fish4Knowledge Deliverable D7.5

Year 2 annual report to EC - Core

Principal Author: UEDIN
Contributors: All partners
Dissemination: PU

1 Project Objectives for Year 2

The main objectives of project year 2 were to:

- Make video capture and distribution a routine process: achieved.
- Enhance the fish detection and tracking basic algorithms to the point of routine usage: achieved.
- Develop a fish recognition algorithm for the top 10 species, which covers 95% of detected fish with accuracy of 90+%: achieved.
- Acquire ground truth for the fish detection, tracking and recognition: achieved.
- Set up the bulk video and result database storage: achieved.
- Integrate the video capture, result storage and bulk data processing components on the NARL supercomputer: achieved.
- Design the bulk data processing workflow, and a software system architecture that enables it: achieved.
- Construct the workflow system architecture for the on-demand data processing and delivery: in progress.
- Design an interface that would enable the marine biologists to ask relevant questions: in progress.

As an indication of progress, we present here a summary of the quantity of data captured and processed as related to the project years 1 and 2, plus the data acquired from before the project started:

	Month	VR	HVR	VDT	VR	FD	FT	FR
< 2010	All	9848	~ 1641	–	–	–	–	–
2010	All	168998	~ 28166	2157	1916	14435042	1654896	1118527
2011								
	1	26899	~ 4483	1492	319	2515011	315750	28885
	2	23363	~ 3894	1183	297	1981170	265403	41357
	3	33233	~ 5539	1512	471	3102489	328759	73535
	4	33010	~ 5502	3598	2504	11747293	1237417	880299
	5	24843	~ 4140	949	–	1431802	160529	–
	6	23974	~ 3996	982	–	1793251	201912	–
	7	29711	~ 4952	745	–	2065374	204041	–
	8	28625	~ 4771	469	–	1043464	91730	–
	9	23782	~ 3964	395	–	682116	58230	–
	10	16943	~ 2824	197	–	456466	36420	–
	11	13164	~ 2194	106	–	222536	17951	–
	12	19182	~ 3197	77	–	224198	15522	–
Total 2011		296729	~ 49455	11705	3591	27269769	2933664	1024076
2012								
	1	16437	~ 2739	178	118	643487	42352	24382
	2	11323	~ 1887	6	2	34053	2467	90
	3	8976	~ 1496	6	–	38171	2779	–
	4	7592	~ 1265	4	–	66187	6799	–
	5	8402	~ 1400	4	–	14354	1263	–
	6	8623	~ 1437	8	–	58845	3366	–
	7	8651	~ 1442	5	–	20412	1677	–
	8	6178	~ 1030	–	–	–	–	–
	9	8536	~ 1423	–	–	–	–	–
Total 2012		84718	~ 14120	212	120	876850	60766	24472
Total 2010-2012		550445	~ 91741	14074	5627	42581661	4649326	2167075

Table 1: Number of: recorded videos (VR), hours of video recorded (HVR), videos with detection and tracking done (VDT), videos with recognition done (VR), fish detected (FD), fish tracked (FT) and recognized (FR) in the time period 2010-2012 for all cameras

The main objectives for year 3 will be to:

- Enhance the detection and tracking algorithms.
- Extend the species recognition algorithm to more species and higher accuracy.
- Complete system integration (workflow and user interface)
- Evaluate system performance
- Enhance system to increase data analysis and query answering speed
- Evaluate usability by marine biologists
- Catch up with all previously recorded videos.

2 Work Progress and Achievements during the Period

2.1 WP 1: Video Data Analysis

The aim of WP1 is to detect, track and recognise fish in underwater videos. At the end of year 2, we have developed reliable methods for detecting, tracking and recognising fish, and these modules are now being used for the higher-level analyses such as behaviour understanding and event detection but also for the annotation tasks, the population statistics generation and the workflow composition. In detail, the main activities in Year 2 for WP1 have been: 1) to improve the performance of the fish detectors and trackers developed in Year 1, 2) to identify descriptors for supporting both the previous tasks and the recognition task and the behaviour understanding one. 3) to build robust classifiers for identifying as many fish species as possible and 4) to develop methods for supporting the annotation and labelling tasks.

2.1.1 T1.1 - Fish detection

In order to deal with the peculiarities of the underwater domain (such as light changes, murky water, waving plants), we first developed four background-modelling approaches (see Deliverable 1.1) which dealt with the presence of periodic and multimodal backgrounds, illumination variations and arbitrary changes in the observed scene: the Adaptive Gaussian Mixture Model ($AGMM$) [19], the Adaptive Poisson Mixture Model ($APMM$) [8], the Intrinsic Model (IM)[12] and the Wave-Back (WB) [11] approach. The downside of these approaches is that they rely on the "wrong" assumptions that the background happens more often that the foreground and that the statistical distribution of background pixels is gaussian, when it has been demonstrated that natural images exhibit non-gaussian statistics [18]. These considerations have an impact on: 1) how to model foreground/background pixels and 2) how to update the background model. To deal with these aspects, first, a variant of the original codebook approach [9] has been adopted, where each background pixel is described with a codebook composed by codewords comprising colors transformed by a color distortion metric. However, the Codebook has shown many limitations with videos 320×240 at 5 fps, because it requires, in the training phase, a long sequence of "stable" background images.
Thus, we have also proposed an approach that models the background pixels with a set of

neighborhood samples instead of with an explicit pixel model [1]. In detail, we model the background with a set of N previous samples:

$$M(p_B) = \{V(p_1), V(p_2), \ldots V(p_N)\}$$

where V is the representation of the pixels in given color spaces (in our case Lab, HSV, RGB and YIQ color spaces). A pixel p_X is classified as background if:

$$\{S_R(p_X) \cap M(p_B)\} \geq T$$

where $S_R(p_X)$ is a hypersphere with p_X as center, and, T is a fixed threshold. The background update mechanism is based on a simple strategy that does not replace the oldest values first, but the pixels to be replaced in the model are identified randomly according to an uniform pdf. This approach demonstrated to be very effective in detecting fish achieving an average detection rate of 75% on the ground truth shown in Fig 1.

The main problem of all the above methods is that they perform well only when faced with specific effects in the scene, e.g. the Wave-Back algorithm [11] usually works fine in the case of repetitive scenes and with low-contrast colors but it is weakest when object erratic movement and sudden lighting transitions are present, whereas mixture of probability density function based models are able to model multimodal backgrounds but they ignore the temporal correlation of color values.

For this reason, we adopted Adaboost for its generalization capability [13]. In detail, the training process in Adaboost consists in building a binary classifier (0,1) using a set of weak classifiers:

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

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 minimizing the error related criterion ϵ [13]:

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

with D_i and $y_i \in [0, 1]$ are, respectively, the error distribution and output of the classifier c_t at the i^{th} iteration. According to the previous considerations, we have used six background subtraction based approaches as weak classifiers, namely, Adaptive Gaussian Mixture Model ($AGMM$), Adaptive Poisson Mixture Model ($APMM$), Intrinsic Model (IM), Wave-Back (WB), Codebook and the new approach above described.

Before applying Adaboost, a parameter optimization phase of the detection algorithms was carried out. More specifically, we performed a parameter-space exploration in order to optimize the performance in terms of detection and segmentation accuracy. However, three main problems were identified during the optimization phase: 1) most parameters are continuous-valued, so several values have to be sampled for each parameter, 2) the number of tests increases exponentially with the number of parameters and values of each parameter, and 3) it is important that the test cases be designed in a way to discover dependencies between parameter values.

The algorithm we applied to build the set of test cases resorts to combinatorial designs, well-established in software testing [7], and has been devised to find the least number of test cases such that each pair of values from different parameters is covered in at least one test set. An example is shown in Table 2 where we have three parameters and for each one three values.

Video	Resol.	Objects	Detections	Features									
				C	**DB**	**DI**	**LC**	**PM**	**FCB**	**SPC**	**A**	**WC**	**AC**
1	L	97	1058	–	–	X	–	–	X	–	–	–	–
2	L	525	8925	X	X	–	–	X	–	–	–	–	–
3	L	321	6420	X	–	–	X	X	X	X	–	–	–
4	L	126	1284	–	–	–	X	–	X	–	–	X	X
5	L	147	3072	–	–	–	X	–	X	X	–	–	–
6	L	959	16321	X	X	–	X	X	X	X	X	X	–
7	L	93	1927	–	X	X	X	–	–	–	–	–	–
Total		2268	39007										

C = Crowded **DB** = Dynamic Background **DI** = Dynamic Illumination **LC** = Low Contrast **PM** = Plant Movements
FBC= Fish with Back. Color **SPC** = Stripped Color Pattern **A** = Algae on Lens **WC** = Water Clearness **AC** = Extreme Atmospherical Conditions

Figure 1: Ground Truth Dataset on Videos 320×240 5 fps

Trying all the possible configurations implies to test $3^3 = 27$ cases, whereas with our approach only 10 test cases are generated. For example, the test case (1, 20, 300) is not included since the pair (20, 300) is already considered in the test cases. It can be proved that with the proposed approach the number of test cases increases logarithmically with the number of parameters and quadratically with the number of values per parameter.

Parameters		
A	**B**	**C**
1	10	100
2	20	200
3	30	300

Test cases		
A	**B**	**C**
1	10	100
1	20	200
1	30	300
2	20	100
3	30	100
2	10	200
3	10	300
2	20	300
2	30	200
3	20	200

Table 2: An example of the combinatorial approach for the selection of the test cases in the parameter optimization phase

Table 3 shows, for each algorithm, the number of parameters, the number of total possible values across all parameters, and the corresponding number of computed test sets.

The achieved results for the best detection approaches and the Adaboost one are shown in Fig. 2.

We also added a post-processing level based on Markov Random Fields to improve robustness of detectors to noise, reduce the effects of morphological filters on the detected blobs and to respect object boundaries. Markov Random Fields adopt a global inference using local information for edge-preserving and for an optimal partition "background-foreground" of the

Algorithm	*Num. parameters*	*Num. total values*	*Test cases*
APMM	4	27	42
Codebook	2	11	36
AGMM	6	40	50
IM	2	14	36
WB	1	6	6
The New Algorithm	2	12	36

Table 3: Parameters and total number of values tested for each detection algorithm

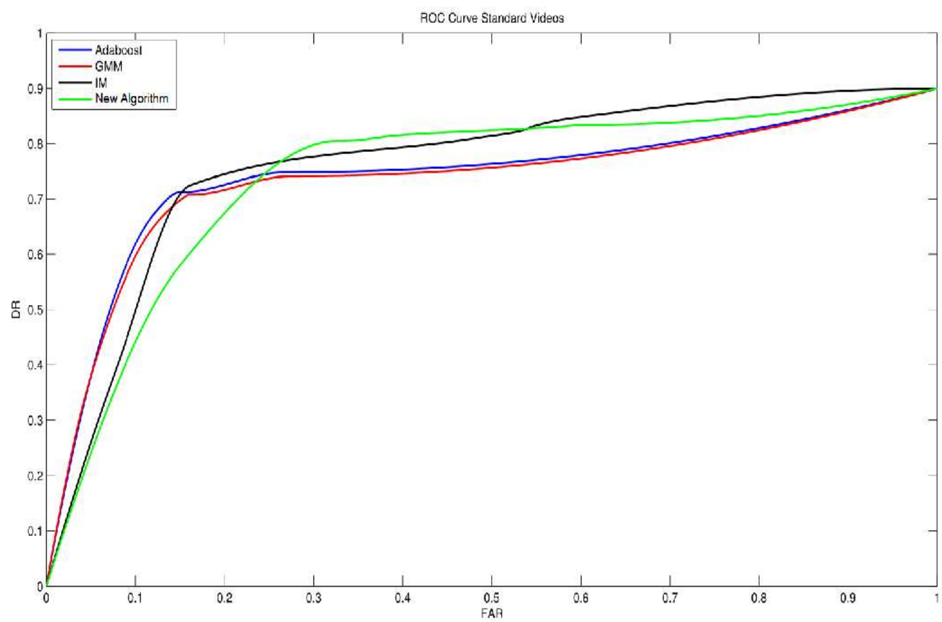


Figure 2: ROC Curve of the best fish detectors and the Adaboost when processing low resolution videos

processed frame. However, the results were not satisfying because the probabilities of pixels to be background or foreground are either one or zero (the employed detection algorithms provide as output binary masks).
Although, as shown in Fig. 2, the results in terms of detection rate can be considered good, the false positive problem is still unsolved. This had led us to the add a side processing level which assigns a score to each detected blob according to the some specific features as described in the "Fish Description" section.
Finally, in Year 2 the fish detectors were tested only on the lower resolution videos since the higher native spatial resolution ones were not available yet.

The detection performance has been improved by adding a post-processing filtering stage, in which objects are selected according to a confidence level describing how sure we are that a detected blob be a fish. The computing of this score is based on the analysis of the color and motion vector in the proximity of an objects 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). A few examples of detection scores is shown in Fig. 1. The inclusion of SIFT key points, extracted from fish clusters, into the computing of the detection confidence level is under investigation so as to reduce the number of false positives.

Although such methods have demonstrated good performance in detecting fish, especially when combined with the post-processing filtering stage (on average, a detection rate of 75% and a false alarm rate of 10% against high quality hand-labelled ground truth), they still present some drawbacks mainly due to the multimodality of the background that cannot be modelled using a specific probability density function, as deviations from the assumed model are ubiquitous. For this reason we are now implementing an approach that does not opt for a particular form of the probability distribution function (pdf), where each background pixel is modelled by a mixture of arbitrary pdf, whose distribution is identified while new pixel values appear, and by a set of samples of values rather than with an explicit pixel model.

2.1.2 T1.2 - Fish tracking

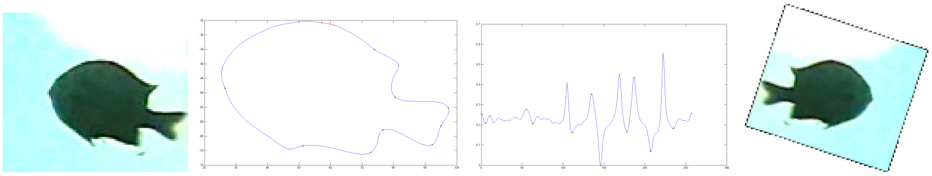
Our tracking algorithm is based on a covariance representation of the object's model, as firstly introduced in [10]. This approach proved to be able to model non-rigid objects and to capture its spatial and statistical properties as well as their correlation within the same representation. Each object in the scene is modelled by the covariance matrix of a set of pixel-based features extracted from the latest object's window. The following features are extracted from each pixel: location (x-y coordinates), colour (RGB and hue channels) and local histogram (mean and variance of a 5×5-window histogram centered on the target pixels). After the feature image for the object's region is extracted, the covariance matrix of these feature vectors is computed, and used to model the current instance of the object. However, in order to take the objects history into consideration, its actual model is computed as the mean of the covariance matrices computed in its most recent appearances. Particular care has to be given to the 'mean' operator, since covariance matrices do not lie in a Euclidean space (e.g. their element-by-element sum may not be a valid covariance matrix). For our purposes, a Lie algebra update mechanism has been adopted to be able to merge several covariance matrices. Once the object model has been computed, it can be used to locate the position of the object in the new frame. One of the most delicate steps in the algorithm is the definition of the object's search area, given its latest-known location: if the search area is too large, the object might be mistaken for a different one passing

by the same region; if it is too small, a sudden acceleration might get the object out of the search area, thus causing the tracker to lose it. Our approach for the computation of the search area for an object is based on estimating its expected movement as a weighted combination between its average speed (i.e. the distance by which it moves between two consecutive frames) and the average speed of objects having a similar size (used to initialize the motion model). Given the expected maximum movement M, the search area will be a circle centered at the object's current location with radius equal to $M + s/2$, where s is the maximum between the object's height and width. Further modifications to the search area apply if the object has been missing for one or more frames. In this case, the search area will be further expanded proportionally to the number of missing frames, and in the estimated direction that the object might have been moving to, according to its history. The current location of the object is then determined by comparing its covariance model with the covariance matrices computed from candidate windows within the object's search area. As for the mean, it is not possible to use a subtraction-based metric to compute the distance between two covariance matrices. To overcome this limitation, Forstner's distance, using generalized eigenvalues, is applied.
The developed object tracker is very promising and the results show performance of about 90% (when compared with the ground truth shown in Fig. 1) in following fish trajectories even in the case of multi object occlusions. A detailed performance evaluation of the fish tracking approach can be found in [17]. Since testing tracking algorithms is a not trivial task, we have also developed an on-line method for performance evaluation which does not use any ground-truth data. It specifically analyses the regularity of motion, shape and appearance (see Deliverable 1.1) of each tracking decision and combines this information through a naive Bayesian classifier, obtaining a probability score representing the overall evaluation of that tracking decision. The results (see [16]) show how the proposed approach is able to reflect the performance of tracking algorithms on different target motion patterns.

2.1.3 T1.3 - Fish description

Fish description in the Fish4Knowledge project is necessary to support fish detection, recognition and behaviour understanding. In detail, different features have been used/conceived to meet the following three goals:

- to discriminate fish (as coming out from the foreground identification process) from other background objects in order to reduce the number of false positives due to errors during the detection process. To achieve this goal, we adopted a set of specific features of real-world objects. The set of considered features exploit two main concepts: 1) the "human perceptual organization model" [6] to discriminate blobs that are most likely produced by the motion of a biological object from blobs that may arise due to changes in the background (i.e. luminosity), 2) the "motion objectness" to compute the probability that a change detected by the above algorithms is due to fish movement instead of background movement (e.g. corals or algae). The feature vector, containing the above objectness's measures and the perceptual organization energy value of each detected blob, is then given as input to a naive Bayes classifier with two classes: *"object of interest" (OI)* and *"false positive" (FP)*, which computes the probability that the considered blob is a fish or not. A detailed performance evaluation of the proposed approach can be found in [14], however, by filtering out all the blobs with estimated probability lower than a threshold, we were able to reduce the number of false positives to about 10% with a detection rate



(a) . (b) . (c) . (d) .

Figure 3: Fish orientation demonstration: (a) original fish image; (b) fish boundary after gaussian filter; (c) curvature along fish boundary; (d) oriented fish image.

of 80%..

- to recognise fish species and to cluster together fish images. In these cases, the most important used descriptions are the color, the texture and the contour of the fish. To compute these features for the recognition, the following steps are taken: Firstly, the Grabcut algorithm is employed to segment fish from the background, based on the obtained contour from the detection method. Secondly, we propose a streamline hypothesis, which uses the assumption that the head is smoother than the tail. The fish orientation is calculated by weighting each contour pixel with its local curvature scale, and this algorithm is used to align all fish horizontally where the head of the fish is located on the right. Based on correctly aligned fish, further features like the color and texture of the fish are extracted and used both in the fish recognition and clustering methods. The alignment procedure in shown in Figure 3. Evaluation using 1000 ground truth images shows 95% of the detected fish are correctly aligned (within 20 degrees of ground truth).
- to model and recognise fish behaviour and its interaction with the surrounding context. In our case we have employed trajectory features to model and describe fish behaviour. In detail, we represent each fish trajectory $T = \{(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)\}$ (i.e. the sequence of centroid coordinates provided by the tracking algorithm) with an HMM, whose output variables are position coordinates, speed and direction of the fish, modelled by mixtures of Gaussians. Differently from the traditional HMM approach, we do not force the states of the model to match real world locations, instead we let the HMM learn its own internal configuration by applying the Baum-Welch algorithm and feeding a trajectory or a set of trajectories as input. Moreover, we do not apply the state transition probability as in [20] because it does not hold for 3D unconstrained motion such as fish movement. All states have the same initial probability.
  This HMM based fish description has been applied both for learning fish-species behaviour (solitary, pairing, etc.) and for detecting uncommon trajectories which may be either actual anomalous fish behaviour to be investigated by marine biologists or errors of the tracker. The HMM based representation of fish trajectories allowed us to reach performance in understanding fish behaviour and in detecting anomalous trajectories of, respectively, about 80% and 86% [15]. However, one of the main limitation to the above approach is the low temporal (only 5 fps) resolution of the processed videos.
  A detailed explanation of the adopted descriptors is given in Deliverable 1.2.

2.1.4 T1.4 - Fish recognition and clustering

For fish recognition, a novel method is proposed to recognize fish in an unrestricted natural environment from underwater videos. A combination of SVM (support vector machine) and Balance-Guaranteed Optimized Tree (BGOT) are used. The Balance-Guaranteed Optimized Tree (BGOT) helps to resolve the error accumulation issue for tree classification and makes use of the inner-class similarities among fish species. The one-vs-one SVM classifier is used at each of the nodes in the tree to separate the different species into subclasses. The framework is illustrated in Fig 4. We compare the hierarchical classification against the Ada-boost method (75.3% AR) and flat SVM classifier (86.3% AR). The automatically generated hierarchical tree (BGOT), which chooses the best splitting by exhaustively searching all possible combinations while remaining balanced, achieves an AR of (90.0%), which is significantly better than Ada-boost and the flat SVM classifier. This result is over 10 species and based on 3179 ground-truth images. Although we expect that there will eventually be about 100 identifiable species, it seems to be the case so far that these 10 species cover more than 95% of the detected fish. We are currently extending the recognition to more than 20 species and also adding a "null" class detector.

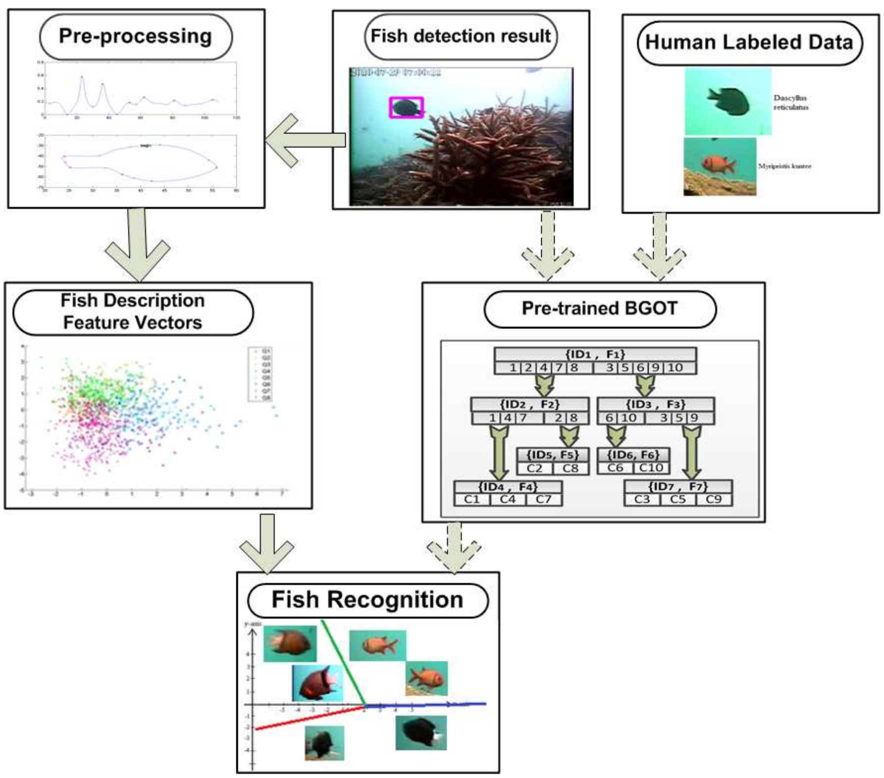
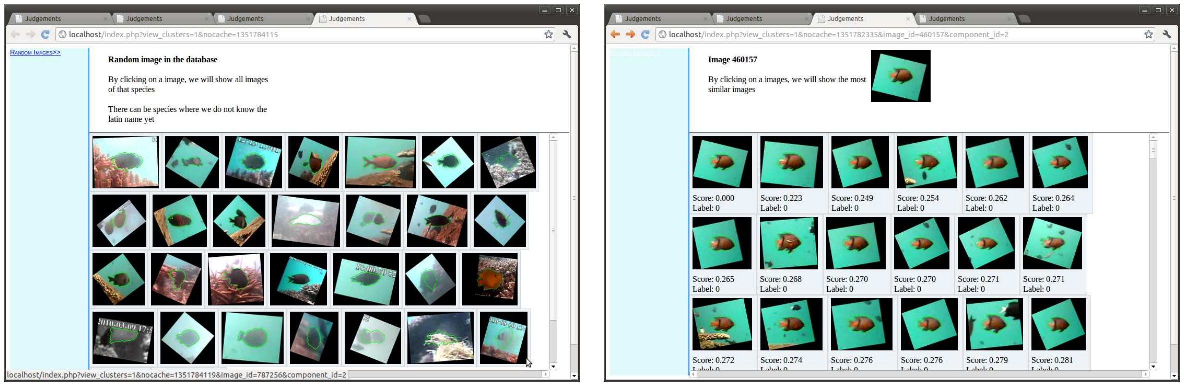


Figure 4: The framework of our BGOT-based hierarchical classification system. The work flow of dotted arrows shows the training procedure and the solid arrows indicate the recognition procedure.

For fish clustering, the Information Bottleneck is proposed as a distance measure between low resolution fish images in order to perform image retrieval, where the goal is to find relevant images in a large dataset. Current methods for image retrieval either compare histograms of SIFT or feature vectors. The advantage of the Information Bottleneck is that it compares



(a) The first interface allows you to choose a random fish to see if similar fish can be found in the dataset

(b) The second interface put the selected picture of the fish on top and shows in a order way similar fish that in found in the dataset, average query time is around 1 second

Figure 5: Query interfaces

feature sets. Not enough SIFT features can be extracted from low resolution images to create good histograms and creating feature vectors often requires specific domain knowledge. The proposed algorithm allows us to efficiently compare feature sets for a large database of images by converting the feature sets to bit strings, which are comparable with Locality Sensitive Hashing (LSH) on vectors. This allows us to index the images in an efficient manner for the task of image retrieval. Experiments show that this method outperforms both the bag-of-feature approach and LSH with a domain specific feature vector on a database of 20074 fish images. The fish clustering is being used to support the annotation of fish images for recognition, see [5]. A webinterface (Figure 5) has been developed to search a database of 393101 images for nearest neighbors.

2.2 WP 2: Interactive User Query Interface

The work on the Interactive User Query Interface has undergone a change of focus compared with the UI component development plans described in the Fish4Knowledge project proposal. The need for ground truth for the training and evaluation of computer vision components within the project lead to the construction of additional user interface support, which was unforeseen in the original proposal. User interfaces were developed for collecting ground truth from both expert and lay users.

In addition, user requirement studies identified a need to explicitly communicate uncertainty metrics and evaluation results to end users. In D2.1 *User Information Needs* [3], we sketched how the answers to almost all “20 questions” users might ask from the Fish4Knowledge system have associated issues to trust and uncertainty, and we used this to drive the ”Charles” scenario in D2.2 *User Scenarios and Implementation Plan* [2]. For systems that rely on automated analyses, such as that being constructed in the Fish4Knowledge project, trust issues are directly related to the inherent uncertainty introduced by the computer vision components and need to be explicitly addressed.

In D2.3 *Component-based prototypes and evaluation criteria* [4] we identify the types of uncertainty information that need to be communicated to the end user to allow them to understand the relationship between what the system is able to provide and the information needed by the user. In addition, it discusses the quality of the ground truth data obtained with the user interfaces built for this new purpose. The deliverable also gives examples of both basic and more advanced user interfaces that are able to communicate (aspects of) provenance and implicit and explicit uncertainty information, either visually or via an interaction dialogue.

2.2.1 T2.1 - Establish user information needs

The analyses given in the previous annual report are still valid. In this year we have concentrated on establishing sufficient ground truth data to allow realistic assessments of the accuracy of the video component analyses of the captured video data. This is ongoing work, but is sufficient to initiate conversations with marine biologists on high level queries on the data that is now in the system.

The ground truth collection has resulted in insights into the extent to which professional marine biologists are able to identify fish species consistently, with the result that this is not always possible because of both video quality and visual distinctions per species. This has been translated into game-like user interfaces that encourage lay users to participate in identifying fish species that attempts to reach at least the same agreement as the experts, to then be used for increasing the ground truth set available for the video components in the project.

2.2.2 T2.2 - Explore component-based prototypes

During the second year of the project we have re-oriented the emphasis of the user interface development towards obtaining ground truth data from expert and lay users. This has led to the development of interfaces for obtaining ground truth data rather than the Metric Calculator and the Query Engine mentioned in the previous annual report. The Rendering Engine is still a goal of the user interface development, and an initial prototype has already been built as a "strawman" that could be used among the project team to look at the data gathered so far and to discuss how the visual analysis techniques should be presented to end users in the context of fish population metrics. A screen shot of this interface is given in Figure 6.

Having developed this initial prototype, we were able to develop our ideas on the user interface design further.

The goal of the expert annotation is to assign a species name to each of the fish images. Experts are expensive and a scarce resource. We therefore use expert annotators to label only a small subset of our data and developed a cluster-based interface to facilitate their labelling process. The images annotated by the experts can be used not only as training materials for the recognition component, but also as a validation set for the non-expert annotation. We have developed a first prototype for a game-based annotation interface, as shown in figure 7. The game turns the expert-only task of classifying fish species into a much easier task based on visual similarity. The preliminary results from the game-based approach have been promising in terms of user incentives, data quality and learning effects, and the first paper on this topic is under review at the time of writing. More extensive user studies and studies on the game data quality are ongoing.

In addition, we developed an interface to collect ground truth data for fish behavior, using a rule-based interface to collect candidate footage suited for annotation. First experiments to annotate simple behaviors have been carried out using interfaces as shown in figure 8.

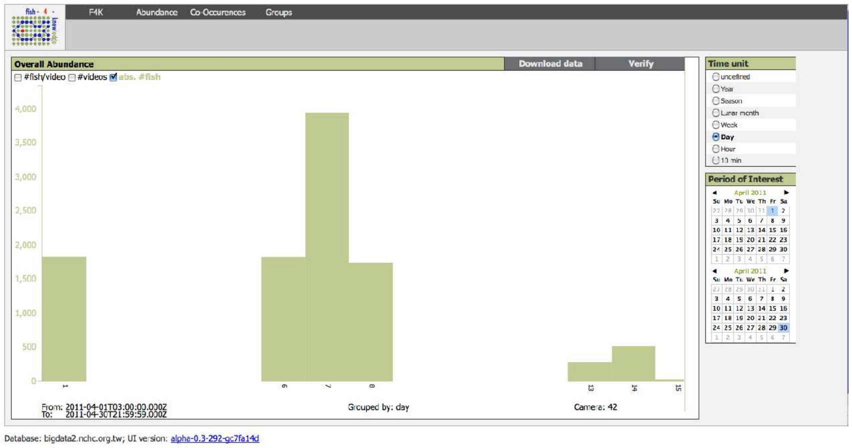


Figure 6: **Initial prototype showing raw counts of fish per video and numbers of videos analysed for the days in April 2011.**

2.2.3 T2.3 - Support for high-level information needs

D2.3 *Component-based prototypes and evaluation criteria* [4] presents a series of mockups that will guide the implementation of the user interface in the third year of the project. These mock-ups give consistent interfaces to tasks the marine biologists will want to carry out, specifically to allow selections of location and period and to obtain analyses of the counts of fish. The system is being built to support these queries and at first sight it is relatively straightforward. The complexity of both the underlying system design and its visualization is in estimating the counts based on the results of the video analysis components and the ground truth data and in conveying these in a way that the marine biologists will trust the results (see example in figure 9). Before we test our designs with marine biology experts we want to be sure we have a system that both works end-to-end and is populated with sufficient examples of video data that more than trivial analyses can be carried out.

2.2.4 T2.4 - End-to-end system integration with data

In order to support an end-to-end user interface that operates on the full dataset, we focus on two data stuctures: a canonical ground truth model and comprehensive summarisation tables. First, we are developing a data model to collect and aggregate all evaluation and ground truth data obtained within the project in a common format. This model allows the evaluation data not only to be used to evaluate the particular component for which the ground truth has been collected, but it also allows the user interface components to use this data to provide approximate confidence values for query results in the user interface targeted at the marine biologists. Second, we use summary tables to be able to quickly answer common queries without going through the big tables on the individual detection level. Unlike the common evaluation table, which only needs to be updated after another run of ground truth collection experiments, the summary tables need to continuously be updated as long as the detection, tracking and recognition components are operational. We assume that with these two data structures in place, we can realize a sufficiently responsive and reliable end-to-end user interface on top of common, off-the-shelf database solutions.

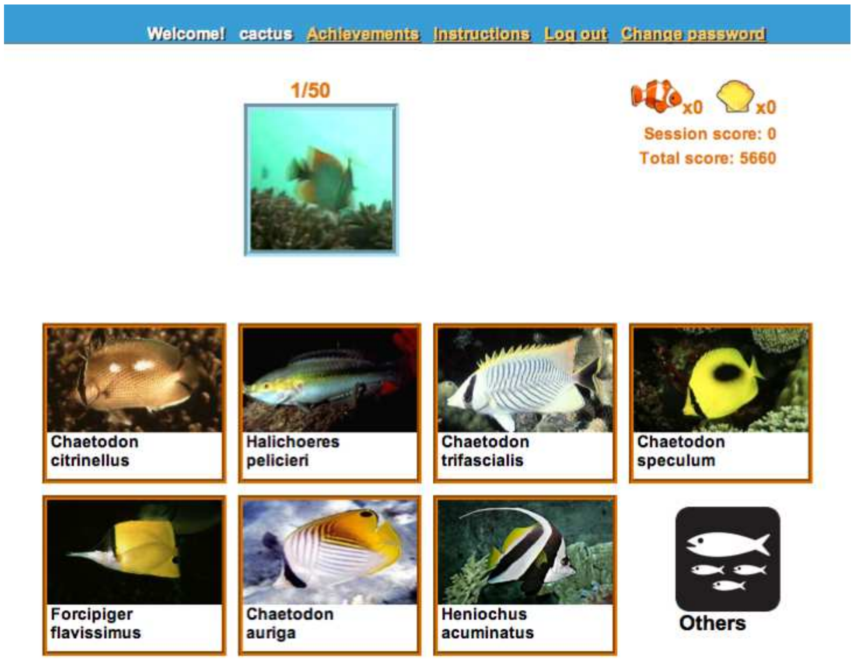


Figure 7: **Game-based ground-truth collection targeting non-expert annotators**

2.2.5 T2.5 - Evaluation and in situational user testing

From the user study we conducted and reported in Deliverables 2.1 and 2.2, we derived 3 primary tasks that underly the data analysis and interpretation:

- A. the identification of **trends** in fish populations;
- B. the identification of **correlations of trends**;
- C. the identification of **levels of confidence** in the identified trends (from task A) and correlations of trends (from task B).

While the marine biologists' main goals are to identify trends and their correlations, they will only be able to do this if they are also able to understand the underlying uncertainties introduced into the system by the automated analysis components. The questions that lead our inquiries are thus the following.

- In order to understand and trust the system, how much knowledge of the computer vision domain do marine biologists need to comprehend?
- What metrics and visualizations are the most understandable for marine biologists to evaluate the levels of confidence in the observed trends and correlations of trends?
- Do the provided metrics and visualizations give sufficient information for marine biologists to derive scientifically valid analyses of the Fish4Knowledge data, including the identification of valid hypotheses and the verification of hypotheses derived from prior knowledge?

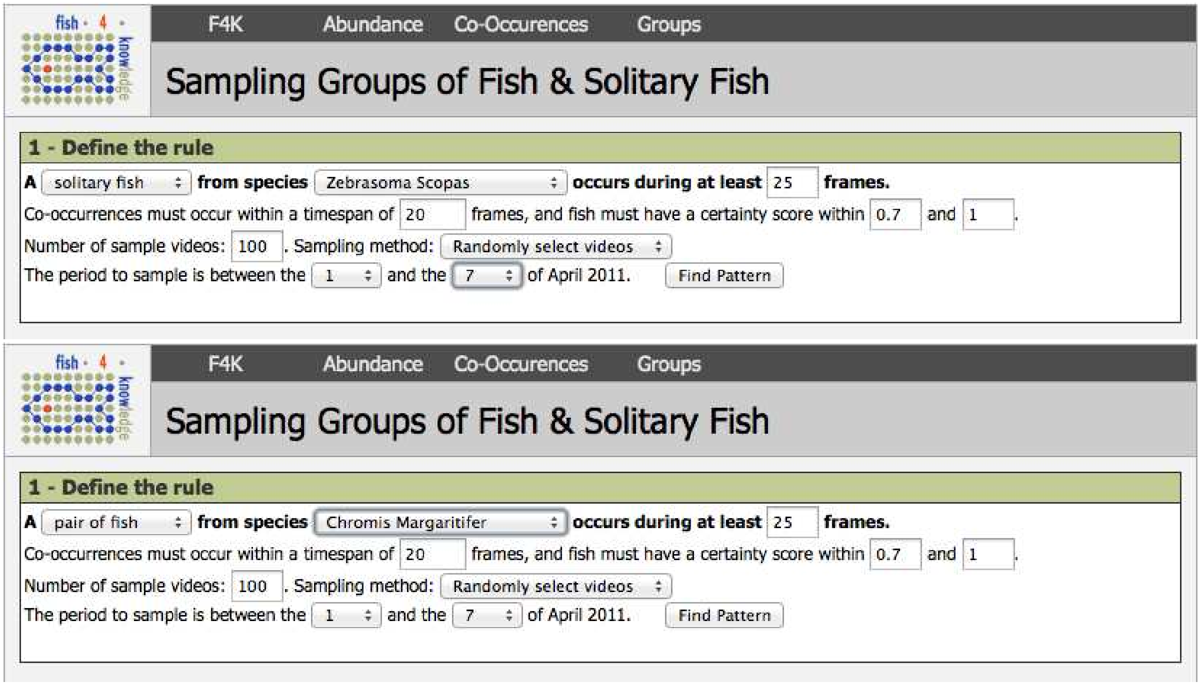


Figure 8: Screenshots of user-defined rules for retrieving solitary and pairing fish.

In order to answer these questions, we will use standard qualitative and quantitative human computer interaction methods. We will start with qualitative investigations to obtain feedback from users using directed tasks with very simple interfaces on a pre-selected portion of the data in the database. As we gain knowledge about the users' understanding of the interpretations of the data in the system we will be able to work in two directions: improve the visualizations of the information (necessary for users to be able to use the system) and, more importantly, understand to what extent users are able to understand and develop some degree of confidence in the statistics that the system is able to supply.

As the system develops, with larger amounts of data and with a more stable prototype interface, we will move towards more quantitative studies to understand better which visualizations are more appropriate for which tasks. These will be developed after gaining understanding of the users' interactions with the system in the qualitative studies.

We are aware that the creation of interfaces to the data analyses in the system is a non-trivial task, requiring different types of expert involvement in both the population and uncertainty metrics. This complexity leads us to anticipate that users of the system will require time to fully understand it, and more time to be able to use it for tasks not pre-specified by ourselves. If the system proves to be sufficiently robust within the lifetime of the project, then we will also carry out longer term studies with a few users to understand how their usage and understanding of the system develops with extended use.

2.2.6 T2.5 - Evaluation and in situ user testing

No action on this task during year 2.

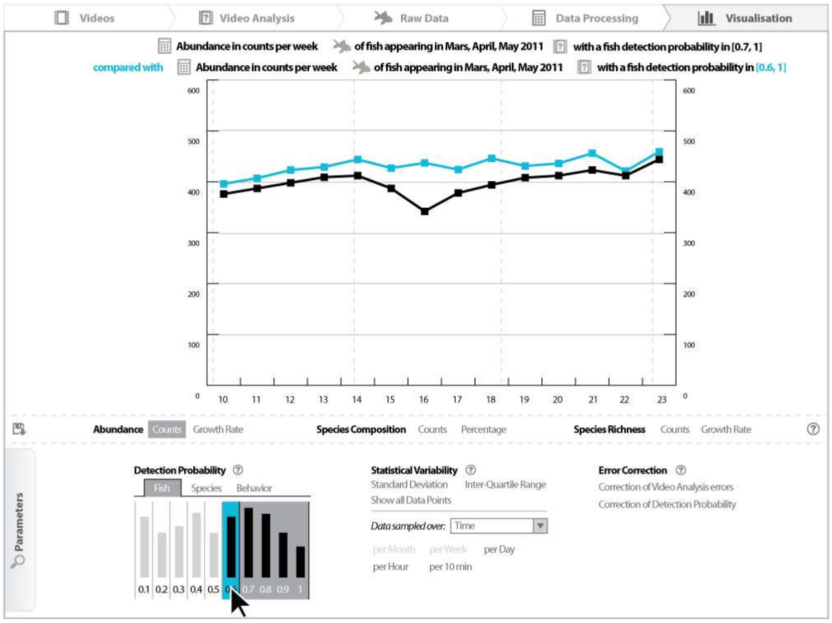


Figure 9: Mockup using explicit *detection probability thresholds*. In this example, the fish detection probability variable is set to the range [0.7, 1]. The user rolls over the 0.6 fish detection probability. It triggers the calculation of a new set of population metrics for a fish detection probability within [0.6, 1]. The new set of population metrics is displayed in blue in the main graph.

2.3 WP 3: Process composition and execution

The workflow component of the F4K project is responsible for the composition and execution of a set of video and image processing (VIP) modules on high performance computing (HPC) machines based on user requirements and descriptions of the video data. It interprets the user requirements as high level VIP tasks, creates workflows based on the procedural constraints of the VIP modules, invokes and manages their execution in the HPC (distributed) environment. More specifically, it takes video data that has been captured by the F4K project partner NCHC NARL, Taiwan and analyses them to answer user queries, by selecting and running a sequence of video and image processing (VIP) modules developed by our project partners in the Edinburgh University, UK and University of Catania, Italy.

During the first year of the project, a set of ontologies that will be used by the virtual workflow machines have been created. This includes an extension of the FAO fisheries ontology, user goals ontology based on user requirements, video description ontology, VIP module capability ontology and their corresponding process library based on broken down VIP procedures. Making use of AI planning technologies, a design of the workflow system and an initial baseline prototype was also developed.

During the second year, workflow efforts have been focused in more detail to deal with the complex management of tasks, including VIP module selection and configuration; run-time job submission, monitoring and progress report; fault detection and handling; and communication and collaboration with other F4K components, in the context of a complex HPC computational environment. During this time, the fundamental make-up of the HPC environment has evolved several times and becomes more powerful to address our project needs, VIP modules have grown and improved, the databases have become more complex and its design and implementation have been coordinated and its use shared among different F4K components, including the workflow system.

To specialise the tasks at hand and cope with changes in the dynamic environment that it is working within, the workflow system has now evolved into a more sophisticated design with three layers: including a workflow process scheduler layer, task generator layer and a Resource Scheduler layer. The workflow system takes user queries, with the assistance of the ontological definitions, it binds them with high level user goals. These user goals are translated into high level VIP tasks that are then broken down into (alternative) lower level tasks that each task may be mapped to one or more VIP modules. Based on ontological definitions, heuristics and user run-time specifications, the workflow machine selects, combines and configures VIP modules to form a ‘working plan’ and send them to the HPC computing facilities via the Resource Scheduler where it prioritises tasks, monitors them and ensure smooth execution. During the execution, databases are used to store partial and final execution results, as well as means to store and track down dynamic execution information thereby enable fault detection and recovery.

To understand and estimate the performance of the workflow machine, the main underlying technologies have been broken down and performance-based experiments have been designed and carried out. Partial results have been obtained and the first comparison studies should be ready in the next workflow team deliverable in January; otherwise, during the next project meeting in April.

The workflow system is a crucial step towards achieving integration in the back end of the F4K system. Its ultimate goal is to provide a scientific workflow method to assist the automatically and efficiently store and analysis of the ‘big data’, as generated by the F4K project, to reliably provide a back-end of a user query portal for marine biologists.

2.3.1 T3.1 - Create Domain Ontologies Based on User Requirements

The design of the ontologies has not undergone any changes since the previous Deliverables (D3.1 and D3.4). A direct impact of the work on interfacing the UN’s FAO fisheries ontology with F4K has resulted in the update of FAO’s Aquatic Sciences and Fisheries Information System (ASFIS) list[1] with 15 of our main fish species in April 2012.

The ontologies will be populated with performance metric information of the software components on the different HPC facilities available to us. These performance metrics will include the average, maximum and minimum execution and queuing times of the software components.

2.3.2 T3.2 - Workflow System Design

The workflow system design has been revised and extended based on year 1’s report. This is to accommodate the user interface needs and the changes in the computing environment.

The workflow manager’s architecture diagram (Figure 10) shows an overview of the components that the workflow interacts with, its main functions, and its sub-components. It interacts with the front end via a server API and updates information in the central database. As can be

[1] http://www.fao.org/fishery/collection/asfis/en

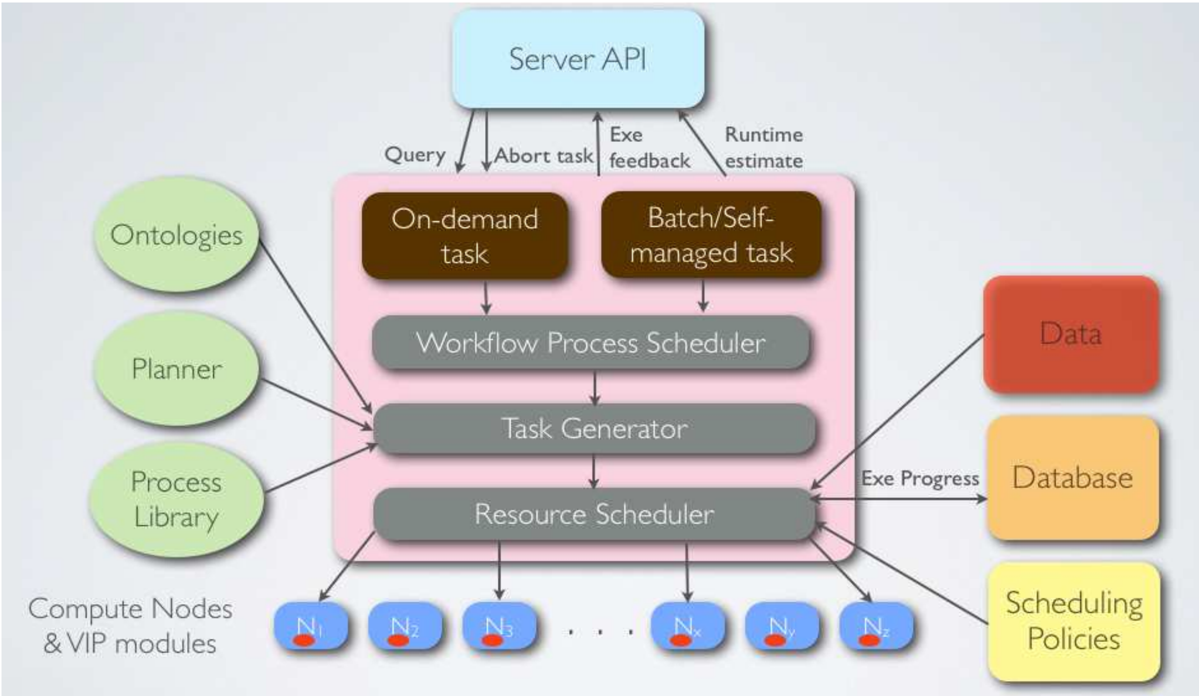


Figure 10: The workflow component binds high level queries from the user interface to low level image processing components via process planning and composition. It also schedules and monitors the execution of the video processing tasks on a high performance computing environment and reports feedback to the user interface component.

seen there are three workflow management sub components: 1) Workflow Process Scheduler; 2) Task Generator and; 3) Resource Scheduler. The main functions supported by the workflow component has been identified as the following with the user interface team:

- Perform on-demand queries (high priority) coming from user.
- Perform batch/self-managed queries (low priority) on new unprocessed videos from NARL's source.
- Perform run-time estimation for a given task when asked by user.
- Update database with progress of execution frequently during execution.
- Stop execution of a task when asked by user (abort).
- Report failure of a task to user (beyond error detection and repair).

The workings of the **Process Scheduler** and **Task Generator** (workflow composition) are described in detail in Deliverable 3.2. The workflow composition makes use of planning and ontologies to identify and set the appropriate parameters to the video and image processing (VIP) modules available on NARL's computing environment. The instance of a selected VIP module along with a set of parameters is a *software component*. The workflow invokes a software component via a **Resource Scheduler**. The next section will outline the development of the intelligent workflow system in line with the computing environment, database and preliminary evaluation.

2.3.3 T3.3 - Intelligent Workflow System

A considerable amount of effort has been dedicated to understanding the computing environment that was made available for our use in order to implement a fully working intelligent workflow system. The workflow mechanism has to adapt to the changes in the heterogeneous computing environment, several revisions to the VIP modules and the development of a middleware dispatcher utility. These have added a level of complexity for the workflow to deal with.

Computing Environment

In our heterogeneous computing environment, we have two distributed platforms, shown in Figure 11 below.

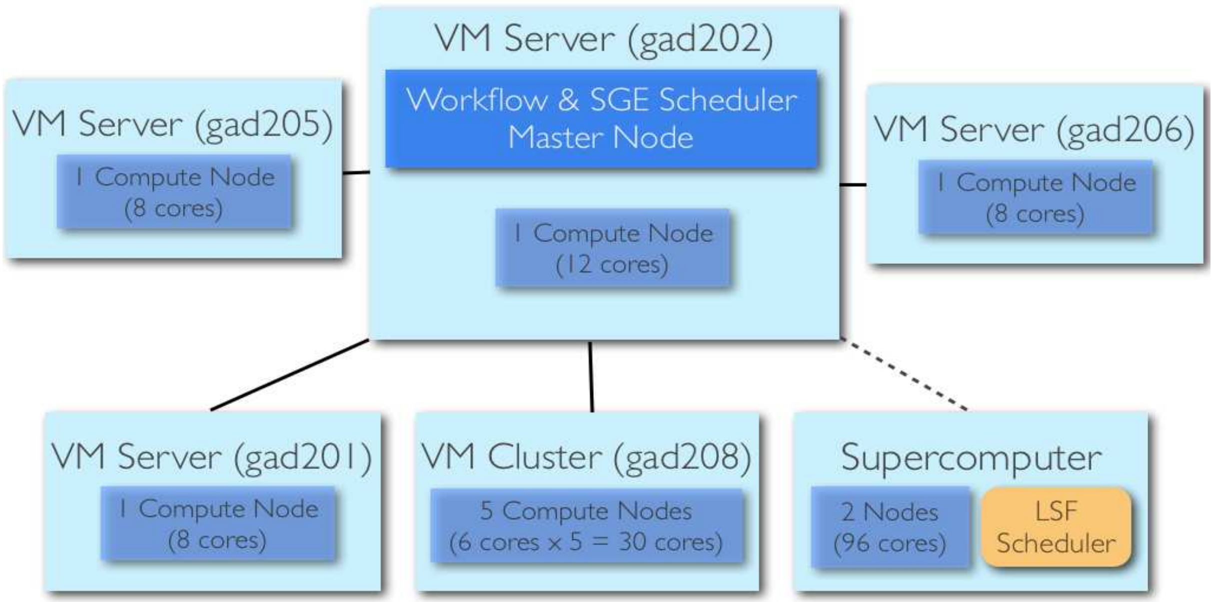


Figure 11: The computing environment for F4K's workflow.

The two main distributed environment are a supercomputer (known as Windrider) and a virtual machine (VM) platform. Two nodes on Windrider containing 48 cores (CPUs) each, totalling to 96 cores are dedicated for F4K use. The resource scheduler on Windrider is the commercial load sharing facility (LSF). On the VM platform, 66 cores are distributed over 9 nodes. The open source gridengine scheduler (SGE) has been set up on the VM platform. In a typical query, potentially thousands of software module invocations (or jobs) are required. This is because thousands of video clips will be needed to process that query, as each video clip only contains 10 minutes' data. A day's processing requires 72 jobs per camera, a week requires 504 jobs, a month requiring 2,160 jobs and a year requiring 26,280 jobs. At present the workflow sends jobs to the SGE scheduler via the DRMAA interface provided as a 'dispatcher' facility by NARL. The dispatcher acts as a bridge to interface the workflow to the resource schedulers in the VM and Windrider platforms. Details on the usage of the workflow computing environment is provided in Deliverable D4.3. Next the monitoring of the jobs using the F4K database is outlined.

Database Design and Implementation

The F4K database is a collaborative effort between the teams to store shared results and information that can be accessed by everyone. Initially it was hosted at University of Catania. However, after several revisions, it was moved to Taiwan, under the hostname *bigdata2.nchc.org.tw*. The revisions allowed all the teams to achieve their desired goals. In addition, the workflow has its own test database, under the hostname *bigdata1.nchc.org.tw* to allow performance testing without hindering the results produced by the VIP teams on *bigdata2*.

Once a job is dispatched for execution, the workflow monitors its execution to ensure that exceptions can be caught and handled appropriately. The workflow primarily uses database tables and log files for the purposes of monitoring. Figure 12 gives a schematic view of the tables used for monitoring.

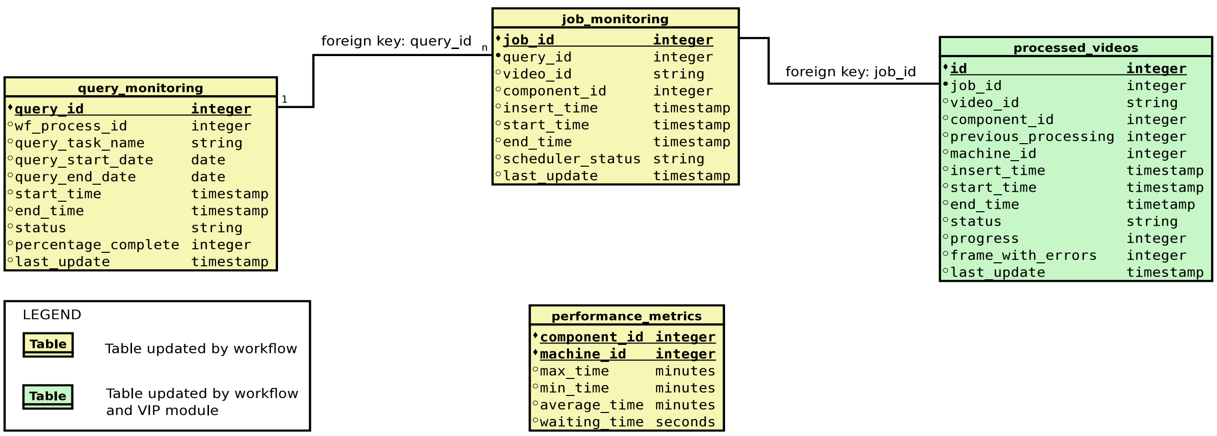


Figure 12: The tables used for query and job monitoring.

The *query_monitoring* table is used to keep track of workflow queries. Consider the query "Count the overall fish population in NPP station from January 1st 2010 to March 31st 2010". The workflow creates one row in *query_monitoring* for this query. Each query in turn can be made up of many jobs. A separate table is populated to keep track of individual jobs, called *job_monitoring*. For this query, 6,480 jobs or rows will be created by the workflow in the *job_monitoring* table. The status of a job is acquired via communication with the scheduler at frequent intervals.

Finally, when a video has been processed (i.e. when a job has been executed), an entry is created in the table *processed_videos*. This table is shared between the workflow and the VIP module that processed the video. The fields "start_time", "end_time", "status", "progress" and "frame_with_errors" are updated by the VIP module. In this way, errors occurring within a VIP module can be detected. Errors related to the scheduler or HPC resource will be reflected in the *job_monitoring* table. The progress, status and execution times for each query and its jobs can be tracked using these three tables. It should be noted that the fields "status", "insert_time", "start_time", "end_time" and "last_update" do not refer to the same value for a corresponding entry across the tables, despite having the same naming convention.

The *performance_metrics* table is used to keep track of the metrics related to the software components. The values aggregated from this table will be updated in the Capability Ontology.

Work in Progress

In order to estimate the performance of the workflow system when it is fully implemented, we first seek to understand the underlying technologies that the workflow has to make use of. These include the resource scheduler, the distributed platforms, the dispatcher utility and the VIP modules. As mentioned earlier we have to cope with two resource schedulers, two distributed platforms and several revisions of VIP modules. At present, these technologies are reaching their stable states with room for minor revisions.

We are currently investigating the workflow's, scheduler's and resources' capabilities using the following criteria:

- Difference in processing times in Windrider and VM platform for fish detection and tracking and fish recognition tasks.
- Difference in the processing times when running a job using varying priority levels (low, medium, high).
- Difference in the processing times when varying the "gaps" between submitting two consecutive jobs.
- When is it necessary to change the priority of a job which is being queued.

For these criteria, we have designed experiments, collected data and will present the partial results that we have obtained at the project meeting in December. The next step is to provide a robust performance based execution and monitoring platform via efficient error detection and handling.

2.4 WP 4: High Performance Storage and Execution Architecture

The goal of WP4 is to maintain a sustainable infrastructure for marine ecology understanding. The infrastructure is composed of networking components: a number (e.g. 10) of video cameras continuously sending the data stream, a massive storage system to store video and processed data, and a high performance computing facility to do data analysis. The core issues to be addressed are how to do fast data query and retrieval with Tera-scale coupled repositories for video data and metadata, and how to accelerate the workflow process execution via compute parallelization. Figure 13 is an overview diagram of infrastructure components.

NARL enhanced video data quality through evaluation of different data frame rates, true pixel sizes, and filters for image quality. NARL also built a high-end tera-scale data storage and a cluster of virtual machines, which serves as a computational node and additionally as a gateway to access backend supercomputers when heavy-duty production cycles are required. To enable the seamless integration of the above computational resources, a job dispatcher was developed to allow the workflow system to submit jobs to different queuing systems.

2.4.1 T4.1 Enhance current video capturing and storage

In response to the requirement of performance evaluation for image processing algorithms, combinations of video source resolution and encoding bitrate under different circumstances (e.g. water turbidity) are recorded, and a better recording scheme is implemented with storage

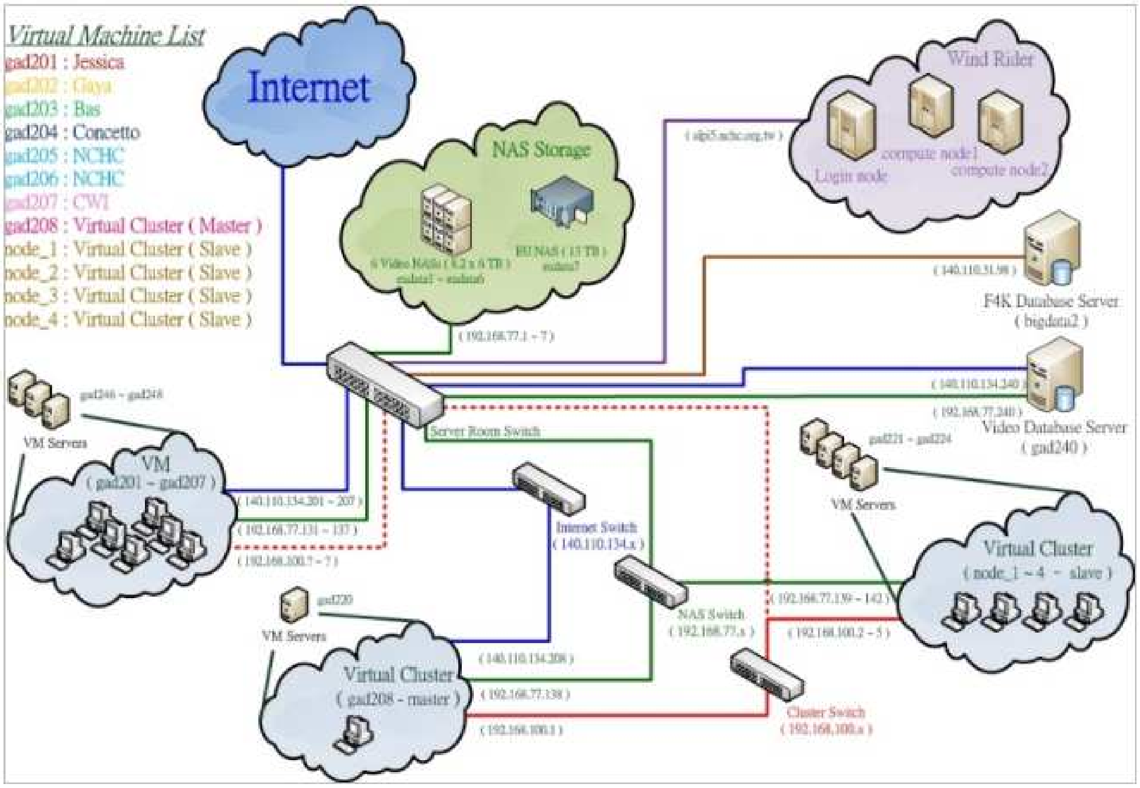


Figure 13: An overview of the hardware architecture available for F4K. The main components include high-performance computing machines, database servers, video storage facilities and web servers.

at the observation site to prevent data loss during transmission over long fat network. For evaluation of 'the-best-can-have' from the actual system, a few hours of raw videos are recorded. The image processing teams are doing ground truth analysis on these raw videos.

Four enhancements have been made to the data capture facilities:

1. Installed a local Processing Server and local NAS Storage at the NPP3 site to store high bitrate video data. This also provides the ability to store raw video data.

2. NARL improved the video quality by implementing new video formats and capturing methods. Video resolution, FPSframes per second and bitrates are increased. The new capturing method provides more stable and accurate video storage. The current capabilities are summarised here:

	CCTV	CCTV	HD
Format	FLV	MPEG4	FLV
Resolution	640x480	640x480	1280x760
FPS	24	24	30
Bitrate	1M	5M	
Capture Method	Stream dump	Stream dump	Stream dump
Site	NPP3, Lan Yu, Hobihu	NPP3	NMMBA

3. The system has been used to capture and store a total of 573,737 videos in video database, of which 41977 were high bitrate videos. The breakdown was:

Format	Bitrate	Site	# of Video in storage	# of video records in database
FLV	200K / 480K / 1M / 2M	All Sites	595465	573,737
MPEG4	5M	NPP3	41,977	none

4. Water temperature and pressure data for the NPP3 site are being recorded for further video analysis.

2.4.2 T4.2 Build data storage facility

The undersea observation project has accumulated 36TB video data at 31/10/2012, and is roughly adding 6TB more to the collection every month as we pushed to the upper limit of resolution. There are also video processing tasks which will generate huge data sets. To accommodate the heavy demand of storage we have procured a rack of storage array, whose capacity scale out to 220TB, and implemented raid 6 to insure data availability. The massive storage system is shared among components on the infrastructure through standard protocols, such as NFS, Samba, etc. The challenge to a scale-out storage system is to allow storage resources growing in-line with data demands as project needs change over time, for example, as the system starts to use high resolution data for precise analysis. This means that the storage system must expand but still maintain functionality and performance as it grows. We've proved the scaling-out capability by plug-in one storage unit, composed by 58 hard disks and capacity up to 125TB, to the infrastructure without interrupting any running process.

NARL upgraded the storage size of video store NAS 1 and NAS 2 from 8.2TB to 14TB and installed 125TB storage for storing video data and F4K data, giving a total of up to 206.5TB of data storage. RAID 6 is being used to improve data protection. A breakdown of the storage is given here:

NAS Storage	Size	Used %	Available %	Comment
NAS 1	14 TB	60 %	40 %	Historical video storage
NAS 2	14 TB	44 %	56 %	
NAS 3	8.2 TB	1 %	99 %	
NAS 4	8.2 TB	25 %	75 %	
NAS 5	8.2 TB	96 %	4 %	
NAS 6	8.2 TB	42 %	58 %	
NAS 7	13 TB	8 %	92 %	VM NFS Shared storage
NPP3 NAS	7.7 TB	100 %	0 %	Storage in NPP3 site
F4K NAS	125 TB	0 %	100 %	F4K data storage
Total Storage	206.5 TB	17 %	83 %	

2.4.3 T4.3 Develop process execution interfaces

In this project, we created two sets of computing platforms to explore a variety of process execution flows. One is a cluster composed of 4+1 nodes of virtual machines and the other is a multi-core supercomputer, nicknamed WindRider, the most powerful computing facility in Taiwan. WindRider consists of 8 compute clusters and one large memory cluster, in total over 25,600 CPU cores, 73,728 GB memory, and 1,074 TB disk. The system offers an aggregate performance over 177 TFLOPS, is ranked 138 among the 500 most powerful commercially available computer systems known to the world in Nov. 2012. Two compute nodes of WindRider are dedicated to F4K project, which can recruit more nodes if necessary. Platform specifications are summarized as following:

VM cluster: 4+1 nodes, each node with Intel i7-2600@3.4GHz CPU, 6 CPU cores, 8GB RAM, gigabit Ethernet link to massive storage. Resources are managed by Grid Engine. Both interactive and batch jobs are acceptable.

WindRider 2 nodes dedicated, each with AMD Opteron 6174 2.2 GHz CPU, 48 CPU cores, 128GB RAM. Resources are managed by Platform LSF. Batch jobs only.

Given these two platforms have a different resource scheduler, an interface to bridge the two schedulers is required so users can compose and submit jobs based on computation requirements without knowing the details of the schedulers. A middleware, Job Dispatcher, was developed to direct the workflow engine to submit jobs to the proper platform and tracks status. Figure 14 describes the logic of the Job Dispatcher.

NARL created a set of specialised virtual machines (VM) for UEDIN, CWI and UCATANIA to allow easy acceess for accessing NARL's computing and storage facilities. VNC and SSH access methods were developed to allow access to the VM. This created these virtual machines:

Server name	gad246a	gad246b	gad246c	gad246	gad247
VN name	VM1 (Jessica)	VM3 (Bas)	VM4 (Concetto)	VM5 (CWI)	VM2 (Gaya)
Num CPU	8	8	8	8	12
Memory 16GB	16GB	16GB	16GB	32GB	
VM disk	64GB	64GB	64GB	64GB	64GB

NARL installed OpenCV, FFmpeg, Libav**, MySQL on the computational resources to provide the resources needed for the scientific and database software.

NARL enabled a successful experiment running the fish detection and tracking code on up to 1008 CPU cores, which also identified a SQL database update bottleneck.

2.4.4 T4.4 Develop distributed data and computational methods

In the image processing tasks, the detection component extracts various features from the video data from low-level features like shape, color, texture, and spatial relations and these data are targeted to be stored efficiently into the relational database. In the distributed computing environment, massive amounts of detection tasks are running at same time and these processes need to communicate with the database instantly to retrieve parameters and store results. Large amount of instances access to database at same time can create heavy loading on the server which eventually becomes a bottleneck in the overall workflow. To overcome this issue we adopted a load balancing mechanism which uses a two node master-slave replication cluster and redirect read-write queries to the master and slave separately. Mysql-proxy is used as the redirecting gatekeeper. With this replication cluster + proxy load balancing mechanism we can easily scale out process capability of database server by adding more slaves when necessary.

NARL developed uniform library components that allowed process execution on both the virtual machine (using Grid Engine) and WindRider processor nodes. This allows the WP3 workflow components access to all computing resources by a uniform interface for job submission, job control, and job monitoring. Computing resources can be dynamically allocated according to the demands of the workflow.

The image analysis teams have compiled the detection, tracking, description and recognition modules so that these models can run parallel or distributed, i.e. on different videos on the windrider platform. The platform provided by NARL allows to run these modules on the 96

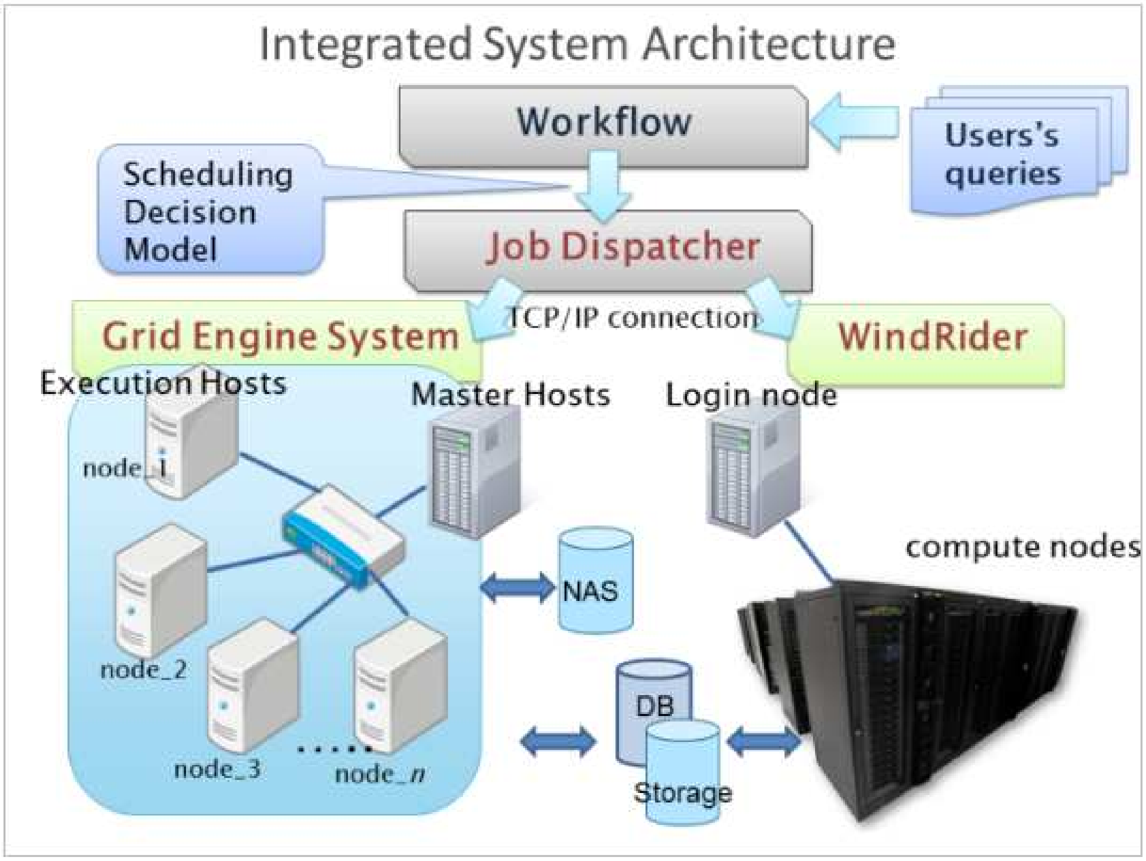


Figure 14: The workflow engine composes job scripts based on users' requirement and calls to Job Dispatcher for instruction on job submission.

CPU windrider machine using the Load Sharing Facility (or simply LSF), which automatically distributes the load over different CPUs and works for both parallel and single CPU modules. Temporary workflow software is created to run both the fish detection and recognition modules using the LSF interface on the bulk of past videos.

At the moment, we have a distributed MySQL server that can deal with the data load from our components. In the near future, we will probably have a separate database for hourly and daily summary tables which provides quick access to the database for most of information requests.

2.4.5 T4.5 Support code parallelisation

Image processing tasks are mostly of the SIMD (single instruction, multiple data) type of computing. They are not well optimized for multi-cores cluster architecture like WindRider. To harvest best performance gains from WindRider, several code parallelization strategies are studied. We've successfully deployed fish detection code on supercomputer with a thousand CPU cores. The experiment ran continuously for 48 hours, and c. 11,000 10-mins video clips were processed during the period of time. Detailed benchmarking data is under evaluation, and also other parallelization strategies. Our goal is to target production runs on at least 1000 cores in parallel during computing intensive periods.

Beside the computing performance, we found some weakness in the data communication between the computing nodes and the database server when a thousand instances are accessing the database at same time, which causes a bottleneck in the process flow. We will investigate better data flow logic to overcome the bottleneck.

2.5 WP 5: System Integration and Evaluation

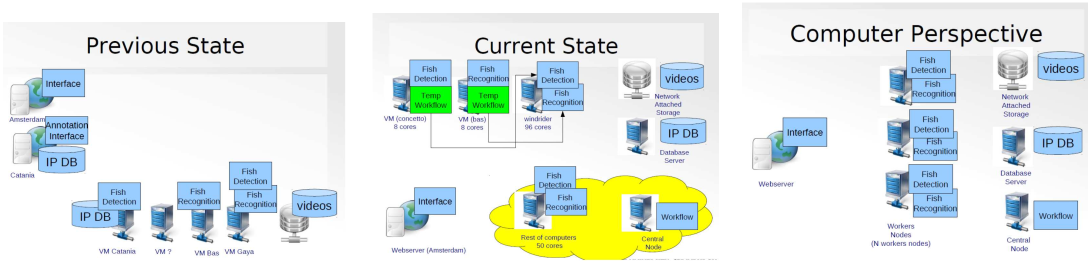
The original system design is described in Deliverable 5.1, where an architecture is proposed in which components communicate with each other using the database. The advantage of this architecture is that each component only depends on access to a database (and access to the storage of the videos). For this architecture to be successful, agreement on the database definition is necessary. These database definition are described and explained in Deliverable 5.2. Due to changes in the definitions, improvements have been performed to the original database design, where this deliverable is kept up to date on our wiki.

In the second year, the focus of the integration has mainly been on getting a working system in Taiwan. Figure 15 show the progress in the integration of the entire system. Last year, the first prototypes of different software components were developed together with the database definition to support these software components, see Figure 15(a). Currently, a MySQL database is used where most programming languages have libraries available to connect to this database package. It also allows all the partners to connect remotely from their institute to databases hosted at other institutes. In the beginning of 2012, the University of Catania hosted the central database, which was moved to NARL, as is shown in Figure 15(b). Afterwards both the fish detection and recognition software was installed on the computer in Taiwan, where the Virtual Machine (VM) allow the image processing groups to experiment with new developed software. The fish detection and recognition software was afterwards installed (and compiled) on Windrider (computer cluster with 96 CPUs). A temporary workflow program is created to run the fish detection and recognition software distributed on the windrider machine.

This created content for the user interface team allowing them to start working with real data. Although the webserver is still hosted by CWI, it is very easy to change this given that the database connection is already working between the interface and database.

A separate test bed has been prepared for workflow, where this currently runs on a cluster of 50 CPUs, where both the first versions of the fish detection and recognition components can be executed. This allows us to develop the workflow component at first in a more controlled environment. The final goal is shown in Figure 15(c), where we are going for a central management structure of the workflow component, but in the development stage, it is good to be able to test and develop each component separately which can be better achieved with our current environment shown in Figure 15(b).

For the evaluation, multiple groundtruth annotation interfaces have been developed in order to obtain data that allows us to evaluate the image processing software. Without this data, the evaluation of the components is impossible, but in most cases obtaining good annotations is difficult. At the moment, we collected around 31,221 annotations for fish detection and 28,264 species labels for fish recognition.



(a) (b) (c)

Figure 15: The progress of the system integration: a) This diagram shows the status of the system at the start of the first year, when everybody was still working on their own prototypes. b) This diagram shows the current state of the system, both the first version of fish detection and recognition components are fully integrated into the system (running on the 96 core windrider machine with temporary schedulers). A test bed is created for the workflow on the fish4knowledge machines, the webserver is still at CWI, but it is able to connect to the database server at NCHC. c) This diagram shows the desired final state of the system. As we still have over a year to achieve this goal, the progress shown in the previous diagrams shows that this should be feasible.

In this section, a short overview is given of the different components for annotating the data:

1. Perla (fish detection): This is a web interface for labelling the contour and trajectory of the fish in the video. An example of this webinterface is shown in the top of Figure 16. It allows multiple people to annotate the trajectory and the contour of the fish and later combine those annotations.

2. Fish game (fish detection): The fish game (middle-left of Figure 16) is a fun way to perform the annotation of fish, where the annotator plays a diver in the game with a camera that has to take pictures of the fish. These picture allow us to define the location of the fish in the video. Notice however that these annotations do not give a contour.

3. Fish behaviour (fish behaviour): For the fish behaviour, an annotation website (middle-right of Figure 16) is created which allows users to search for combinations of species in the videos, for instance if two clown fish appear in the video around the same time. Afterward, we can annotate if these fish are interacting with each other in certain way, for instance pairing.

4. Clustering interface (fish recognition): A website (Figure 17) is created to annotate the fish species, where (left) we first remove the species that are incorrectly classified to that cluster and afterwards (right) link this cluster to a certain species. This allows users to annotate fish images $3\times$ faster than annotating each image separately. It even makes the annotation task simpler so no domain knowledge is required.

5. Species Verification (fish recognition): There are around the 2894 unique fish species in Taiwan, so can users find out to which species an unknown fish in the videos belong. This interface checks the ability of user to perform this annotation task.

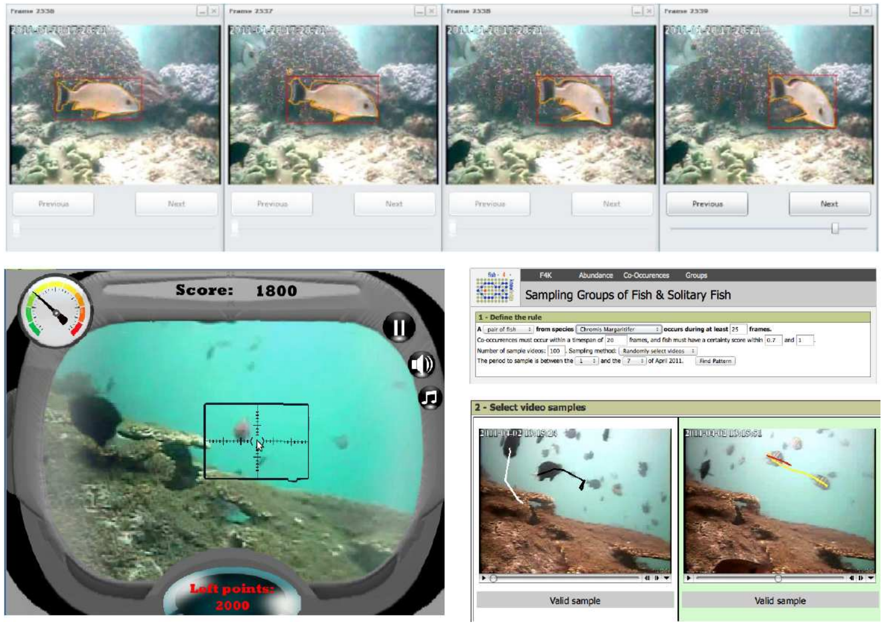


Figure 16: Interfaces that have been developed for annotation of fish detection, tracking and behaviour groundtruth data

2.5.1 T5.1 - Define component interfaces

The different teams have a lot of freedom in how to create their own component. However, in order that the workflow team can easily start and execute the different components, we are working on some guidelines for the executables (the current guidelines are given below):

1. Compatible with updated DB definitions (e.g. video id being a string and any other schema changes announced at most recent F4K meeting).

2. Checks for video file existence in ∼/work/video_dir/video_<uuid>.flv. Otherwise saves (downloads) video to ∼/work/video_dir/video_<uuid>.flv

3. Generates log files in ∼/work/vip_logs/<uuid>-<component_id>.log

4. Updates processed_video table fields (start_time, end_time, progress, frames_with_errors and status). Workflow will update video_id, component_id, previous_processing, machine_id and insert_time. Status field will be pending by default when a new record is inserted.

5. Provides a Readme file with the following (see template below for example)

   - parameter list and description
   - a sample run command
   - dependencies with other modules if any
   - side effects (does it fail if rerun with same parameters?)
   - DB tables updated or affected
   - DB tables accessed (optional)

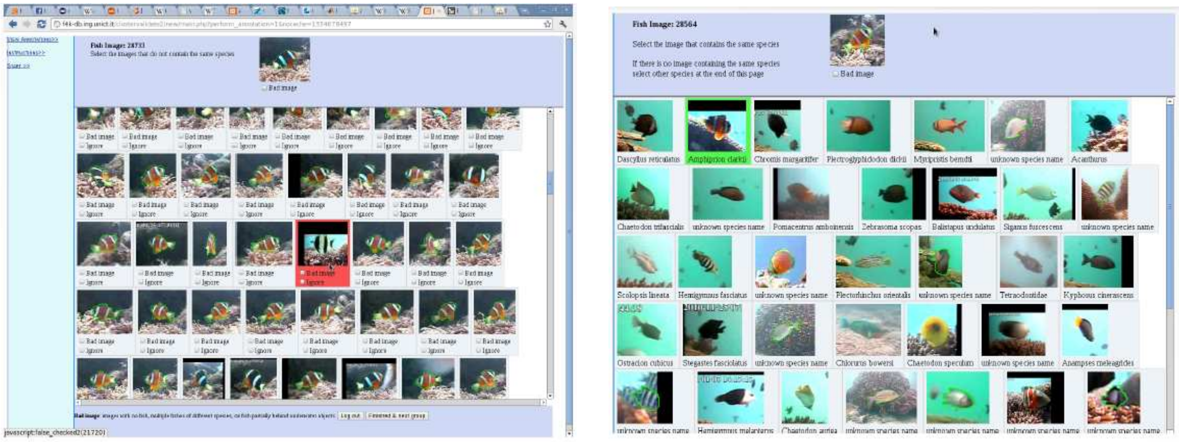


Figure 17: Interfaces developed for the species groundtruth annotation. Left: cleaning a cluster. Right: linking clusters to species.

6. Provides a database configuration file (see below).

7. The current stable version (v1.0) executables should be saved in the relevant directories (e.g. ∼/components/detect_track_fish/v1.0/, ∼/components/recognise_fish/v1.0/ and /components/cluster_fish/v1.0/)

2.5.2 T5.2 - Integration and evaluation planning

For the integration, we have updated the database definitions when moving the database to the servers in Taiwan. The current state of the system is shown in Figure 15(b), which shows that the connection between most components are in place. The fish detection has already detected over 42 million fish images and the fish recognition which is dependent on the fish detection processed around 2 million fish afterwards. The user interface is able to query these results from the database in Taiwan. The workflow component is able to start both the fish detection and the fish recognition software (however this can currently only run on our own cluster machine).

The new challenge in the integration is to communicate the certainty of detecting and recognising fish in the videos. This requires that the image processing modules can be evaluated automatically in the near future. To make this possible, the groundtruth databases need to be integrated into the already existing processing database. Standardisation makes possible evaluation of fish detection and recognition components, together with an interface that can deal with this kind of data and can communicate this to the users of our system.

The evaluation of the different subsystems is already in progress, where the annotation interfaces are developed to obtain data for evaluation. Most of the image processing modules have groundtruth data in place for the evaluation of these systems. For the evaluation of the entire system, the User Interface is the most important component, which has to be finalised. However, it is also important that there is enough content in the database for biologists to look at and this content needs to be accurate enough so that biologists can trust it. Discussions with both the User Interface team and the marine biologists are needed to find a suitable test date for the first evaluation of the system.

2.5.3 T5.3 - First integration and evaluation phase

We are looking at several marine biologist conference where we hope to either publish our work, give a demonstration of the system or give a workshop about the system:

- EAFE Conference - 15-17 April 2013 - Edinburgh
- International Conference on Marine Science and Aquaculture - 15-16 May 2013 - Amsterdam
- ICBLS 2013 - 27-28 July 2013 - Moscow
- European Marine Biology Symposium - 19-23 August 2013 - Galway, Ireland
- ICCB 2012 - 1-6 December 2013 - Eilat, Israel

2.5.4 T5.4 - Second refinement and evaluation phase

No action on this task during period 1.

2.6 WP 6: Project Dissemination

This section describes our progress so far and future plans regarding project dissemination work as described in WP6.

2.6.1 T6.2 - Scientific workshops

Workshop on Intelligent Workflow, Cloud Computing and Systems

As intelligent workflow machines are one of the primary interests and developments to the F4K project, riding the success of our first workshop, we thought to continue it for a second year. As a result, similarly, we had held a special conference session entitled: Intelligent Workflow, Cloud Computing and Systems as a part of the main conference: International KES Symposium on Agents and Multi-agent Systems Technologies and Applications (KES AMSTA), Dubrovnik, Croatia, June 25-27, 2012. In this document, we will refer to this special session as the Intelligent Workflow session in short.

This invited session is to fulfil our project commitments to hold a special interest scientific workshop in the area of intelligent workflow and high performance computing, e.g. Grid and Cloud computing, and also to generate community interests in these subject areas and to disseminate F4K project results, as appropriate. As a result, we were able to include one F4K research paper in this special session. All papers in this special session are published in Springer-Verlags Lecture Notes in AI, as a part of the LNCs/LNAI series.

KES AMSTA is an international scientific conference for research in the field of multi-agent and distributed systems that is highly relevant to our work in intelligent workflow and Grid and Cloud computing. In addition, the conference interests include knowledge representation and systems, semantics based techniques, ontologies, computational complexity that suits our

Artificial Intelligence work. We therefore thought it is a very appropriate venue to hold our scientific workshop and to attract new interests and collaborative opportunities in our interest areas.

There are four chairs for this invited session this year:

- Chair: Dr. Yun-Heh Chen-Burger, University of Edinburgh, UK (F4K project, Edin. Univ. workflow team leader)
- Co-Chairs:
  - Dr. Fang-Pang Lin, National Center for High-Performance Computing, Taiwan (F4K project, Taiwan team leader)
  - Dr. Ching-Long Yeh, Tatung University, Taiwan
  - Professor Lakhmi Jain, School of Electrical and Information Engineering , University of South Australia, Australia

There were five talks included in this conference session and around 20-25 people in the audience. The audience participation were quite active and supportive. As this session is the second year running, speakers and participants had now getting to know each other better that they went out afterwards for food and drinks. They also spent time together for the remaining of the conference.

One interesting and potential promising follow-up with the Edinburgh workflow team is with a fault tolerance researcher, Dr. Rafael Tolosana-Calasanz, University of Zaragoza, to exchange ideas in fault detection and tolerance in a HPC environment and possible research collaboration in the future.

Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications

The First International Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications (VIGTA'12) (website: *http://vigta2012.dieei.unict.it/*) held in Capri (Italy) on May, 25^{th} 2012 in conjunction with the ACM International Working Conference on Advanced Visual Interfaces (*www.avi2012.it*), aimed at reporting on tools, interfaces and methods able to speed-up the ground truth creation process in computer vision applications by supporting users to build up reliable truths in a reasonable amount of time.

The call for papers attracted 17 papers (10 different countries) from which the program committee selected 6 for oral presentations, 6 for poster presentations and 1 for demo presentation dealing with topics ranging from approaches for the generation of large scale ground truth starting from small datasets to user-oriented tools supporting annotators mainly in the task of object detection, recognition, face detection and image segmentation in still images and in video streams to the integration of computer vision methods and ad-hoc hardware for video annotation.

Dr. Benoit Huet from the Multimedia Communications Department of EURECOM delivered the keynote talk "Multimedia Data Collection using Social Media Analysis". The workshop proceedings are published by the ACM International Conference Proceeding Series published by ACM and are available at *http://dl.acm.org/citation.cfm?id=2304496*. The workshop chairs were: Dr. Concetto Spampinato (University of Catania, Italy), Dr. Bas Boom (University of Edinburgh, UK) and Dr. Jiyin He (CWI, The Netherlands).

Workshop on Multimedia Analysis for Ecological Data

The First ACM International Workshop on Multimedia Analysis for Ecological Data (MAED'12) (website: *http://maed2012.dieei.unict.it/*) was held in Nara (Japan) on Nov, 2^{nd} 2012 in conjunction with the ACM Multimedia Conference, aimed at bringing together practitioners and researchers, both in multimedia and in ecology, to share ideas and experiences in designing and implementing novel multimedia analysis techniques and tools for ecological multimedia content. The program committee selected 12 papers (6 for oral presentations and 6 for poster presentations) which address the following topics: Animal identification and behaviour understanding by mining image and video data, plant identification and classification on still images, classification and characterization of habitats, multimedia data processing for pollution monitoring and ecological multimedia data retrieval.

The workshop keynote, "Multimedia Challenges in Sensing the Environment" was given by Prof. Alan Smeaton. The workshop proceedings are published by the ACM International Conference Proceeding Series and are available at *http://dl.acm.org/citation.cfm?id=2390832*. The workshop chairs were: Dr. Concetto Spampinato (University of Catania, Italy), Dr. Vasileios Mezaris (CERTH-ITI, Greece), Dr. Jacco van Ossenbruggen (CWI, The Netherlands).

Workshop on High Performance Computing in Computer Vision Applications

The special session on **"High Performance Computing in Computer Vision Applications (HPC- CVA)"** was organized as part of the 3rd IEEE International Conference on Image Processing Theory, Tools & Applications (*http://ipta12.ibisc.univ-evry.fr/*), Istanbul, Turkey, October 15^{th}-18^{th}, 2012. This special session, mainly, reports on the most recent approaches for improving the efficiency of computer vision applications by exploiting the potentialities of the Cloud, the GPU and the multicore architectures. The call for papers attracted 19 submissions: only 8 papers were selected by the Program Committee as oral presentations. The proceedings of the conference are published by the IEEE and will be available on-line soon. The special session organizers were Dr. Concetto Spampinato (University of Catania, Italy), Dr. Tommaso Mazza (University of Trento, Italy) and Prof. M. Aldinucci (University of Turin, Italy).

Workshop on Visual Observation and Analysis of Animal and Insect Behavior

The Visual observation and analysis of animal and insect behavior 2012 (VAIB 2012) workshop was held in conjunction with the 21st International Conference on Pattern Recognition (ICPR 2012), Tsukuba, Japan, November 11, 2012.

The issues that the workshop addressed included: detection of living organisms, organism tracking and movement analysis, dynamic shape analysis, classification of different organisms (eg. by species), assessment of organism behavior or behavior changes, size and shape assessment, counting and health monitoring.

These problems can be applied to a variety of species at different sizes, such as fruit and house flies, crickets, cockroaches and other insects, farmed and wild fish, mice and rats, commercial farm animals such as poultry, cows and horses, and wildlife monitoring, etc. One aspect that they all have in common is video data.

24 papers were submitted and 18 were presented. About 30 people attended. The papers and more details can be found at: `homepages.inf.ed.ac.uk/rbf/vaib12.html`. There will also be an associated journal special issue titled "Animal and Insect Behaviour Understanding in Image Sequences" in the EURASIP Journal of Image and Video Processing published by Springer.

Organisation of Journal Special Issues

Three special issues of international journals with impact factor are being organized. All of them have an open call for papers, though authors of the papers presented at the above described workshops have been (or will be) invited to submit an extended and revised versions of their papers.

- The Special Issue **"Methods and Tools for Ground Truth Collection in Multimedia"** of Multimedia Tools and Applications Journal (Springer) will address the development of: multimedia processing methods for supporting automatic ground truth generation, methods and tools for combining and comparing ground truth labelled by multiple users in any field of multimedia where ground truth is required, interfaces (adaptive, proactive, mobile, web-based) for collecting ground truth, methods for data representation and integration, interoperability middleware, features, algorithms, and tools. The papers submission closed on July, 30^{th} 2012. Eleven papers were submitted for review of which four were extended versions of the ones presented at the VIGTA'12 workshop. The first review round was completed on Oct, 31^{st} 2012 and the results are: 9 papers were asked for major revisions while the remaining two rejected. The expected publication date is May 2013. **Guest editors**: Dr. C. Spampinato (University of Catania, Italy), Dr. B. Boom (University of Edinburgh, UK) and Dr. J. He (CWI, The Netherlands).

- The special issue **"Animal and Insect Behaviour Understanding in Image Sequences"** of EURASIP Journal on Image and Video Processing (Springer) aims at reporting on the most recent image and video analysis methods for animal and insect behaviour monitoring and understanding. More specifically, the topics of interest range from living organisms detection, tracking, classification and recognition in image sequences to animal and insect motion and trajectory analysis to categorization and natural scene understanding to ontology for describing animal and insect activities in video content. The call for papers has already been circulated and the foreseen deadline for papers submission is January 15, 2013. **Guest editors**: Dr. Concetto Spampinato (University of Catania, Italy), Dr. G. Farinella (University of Catania, Italy), Dr. B. Boom (University of Edinburgh, UK), Dr. M. Betke (Boston University, USA) and Prof. R. Fisher (University of Edinburgh, UK).

- Special Issue **“Multimedia in Ecology and Environment”** of the Ecological Informatics journal (Elsevier). The aim of this special issue is to stimulate the research community on the most recent methods for the processing, interpretation, and visualization of multimedia data recorded for monitoring ecological systems, with particular attention to animal and plant identification and classification and pollution monitoring. More specifically, the topics of interest are: Ecological Multimedia Content Analysis and Processing, Ecological Multimedia Indexing and Retrieval, Computer Vision for Ecological Video/Image Processing, Animals and Insects Behavior and Event Understanding and Video and Signal Based Surveillance of Ecological Sites. The call for papers has already been circulated and the foreseen deadline for papers submission is next Jan, 15th 2013. **Guest editors**: Dr. Concetto Spampinato (University of Catania, Italy), Prof. Benoit Huet (EURECOM, France), Dr. V. Mezaris (CERTH-ITI, Greece) and Dr. Jacco van Ossenbruggen (CWI, The Netherlands).

2.6.2 T6.3 - Two Web-Mounted User Interfaces

F4K Virtual Aquarium in Second Life

Based on decision taken during the last project review, no further development were made to the F4K Virtual Aquarium, as the building is already relatively developed and in a usable stable state. Although, currently, there is no obvious route as how this building may be exploited and no prompting has been made, 122 visitors have visited this site to date (Nov 20, 2012) and viewed our project efforts since it was firstly built.

2.6.3 T6.4 - Interacting with the Marine Biology Community

Outreach Plan to Marine Scientists

Currently, a list of 19 relevant scientific journals in the subject areas of marine biology, marine life in coral reef, ecology and zoology have been collected, through recommendation and evaluation from Prof. Shao. In addition, a list of 12 research and/or educational societies and two web sites with long lists of mailing lists in the above interested areas, have also been found.

It is our current plan to publish our research papers in the relevant journals and advertise F4K project efforts through the above professional bodies and mailing lists through mail shots. The aim is to initially introduce F4K project efforts and to invite the participation of the use of project results. If suitable partners are identified through this process, potential research and development opportunities may be forged as a result.

It may also be possible to participate relevant conversations that are already held in the relevant communities. Alternatively, social networking mechanisms such as Facebook or Twitter may be created specifically for the F4K project for this purpose, if appropriate.

The AQUACAM Project

The initial discussions between the UCATANIA and UEDIN teams with Dr. Owen Day, who is Head of Communications and Biodiversity and co-director of the CARIBSAVE Partnership, over possible cooperation with the F4K technical capabilities and their Caribbean sealife monitoring project have been finalized in the AQUACAM Project.

The Aquacam Research Program is a 3-year collaboration (2012-2015) between the Fish4Knowledge Research Consortium (F4K) (in this case led by University of Catania, Italy, in collaboration with the University of Edinburgh), the Department of Life Sciences and the Centre for Marine Sciences at the University of the West Indies (UWI-Mona) and The CARIBSAVE Partnership (CARIBSAVE) http://www.caribsave.org/. The goal is to develop a new monitoring system for tropical reef fish, using fixed underwater video cameras and computer vision software that can detect and recognize approximately 40 species of Caribbean fish and estimate their body length. The purpose of this research is to improve the management of tropical reef fisheries and therefore increase the sustainability of livelihoods in coastal communities and the resilience of coral reefs to the impacts of climate change. The Aquacam Research Program involves recruiting one PhD student, based at the Discovery Bay Marine Laboratory, UWI, Jamaica and one assistant researcher at the Department of Computer Engineering and Telecommunication, University of Catania, Italy. The outline plan of the collaboration is one PhD student funded by the University of the West Indies based in Jamaica and the assistant researcher partially funded by Fish4Knowledge in Catania. The topic of the Catania researcher is to apply the existing F4K analysis, but to data acquired from a stereo pair of cameras, to allow exact positions and sizes to be estimated. This will allow investigation of more advanced fish recognition, size and health assessment algorithms. The role of the UWI Phd Student is to develop the protocols to collect measurements, compare various fish monitoring methods in relation to Aquacam method, and undertake cost-benefit analysis of Aquacam systems.

Since April 2012 we had various Skype meetings to finalize the research program. An Agreement of Cooperation between the University of Catania and the Universities of West Indies - Mona Campus has been signed by the Rectors of the two Universities. The kick-off meeting has been scheduled at the beginning of October in Kingston, at UWI Mona Campus. As part of the kick-off activities a dissemination workshop entitled **"Fish and Chips: Computer vision software applications for fish Identification and conservation in Jamaica"** has been organized for October 4. This workshop includes a presentation from Dr. Owen day about the C-Fish Initiative in Jamaica, a presentation about the F4K project, and a technical presentation about computer vision techniques for underwater video analysis (see fliers in Fig. 2.6.3 and 2.6.3). The kick-off visit agenda includes a visit to the Centre for Marine Sciences at Discovery Bay for a field trip to assess best locations for camera placement, and discussion of camera choices and set-up with the whole research team.
The AQUACAM team is as follows:

1. **Dr Owen Day (Project Coordinator)**
   Head of Communications and Biodiversity, The CARIBSAVE Partnership
   Regional Headquarters, Hastings House, Balmoral Gap, Christ Church, Barbados
   Phone: +1 246 426 2042; Direct Tel: +44 7525 487595 ; owen.day@caribsave.org

2. From the F4K Consortium:

   - **Prof. Robert B. Fisher (Fish4Knowledge Coordinator)**
   - **Prof. Daniela Giordano**
   - **Eng. PhD Concetto Spampinato**

3. From the Universities of West Indies:

   Dr. Mona Webber

   Head of Department & Senior Lecturer, Marine Ecology, University of the West Indies, Jamaica, WI

   Tel: (876) 927-1202, 927-2753, 935-8630, 935-8291

   Email: mona.webber@uwimona.edu.jm

4. **Dr. Dayne Buddo**

   Lecturer & Academic Coordinator, Discovery Bay Marine Laboratory and Field Station

   Centre for Marine Sciences, University of the West Indies, Jamaica WI

   Phone: (876) 973-2241 ; Cell: (876) 379-6148; dayne.buddo@uwimona.edu.jm

5. **Dr. Karl Aiken**

   Senior Lecturer, Fisheries Ecology, University of the West Indies, Jamaica, WI

   Phone: (876) 927-1202, 935-8292, 935-8291; Email: karl.aiken@uwimona.edu.jm

References

[1] O. Barnich and M. Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, June 2011.

[2] E. Beauxis-Aussalet and L. Hardman. User scenarios and implementation plan. Technical report, Fish4Knowledge project deliverable D2.2, 2012.

[3] E. Beauxis-Aussalet, L. Hardman, and J. van Ossenbruggen. User information needs. Technical report, Fish4Knowledge project deliverable D2.1, 2011.

[4] E. Beauxis-Aussalet, J. He, C. Spampinato, B. Boom, J. van Ossenbruggen, and L. Hardman. Component-based prototypes and evaluation criteria. Technical report, Fish4Knowledge project deliverable D2.3, 2012.

[5] Bastiaan J. Boom, Phoenix X. Huang, Jiyin He, and Robert B. Fisher. Supporting ground-truth annotation of image datasets using clustering. ICPR 2012, to appear.

[6] Chang Cheng, Andreas Koschan, Chung-Hao Chen, David L Page, and Mongi a Abidi. Outdoor scene image segmentation based on background recognition and perceptual organization. *IEEE transactions on image processing*, 21(3):1007–19, March 2012.

[7] D.M. Cohen, S.R. Dalal, M.L. Fredman, and G.C. Patton. The aetg system: an approach to testing based on combinatorial design. *Software Engineering, IEEE Transactions on*, 23(7):437 –444, jul 1997.

[8] A. Faro, D. Giordano, and C. Spampinato. Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1398 –1412, dec. 2011.

[9] Kyungnam Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 5, pages 3061 – 3064 Vol. 5, oct. 2004.

[10] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 728 – 735, june 2006.

[11] Fatih Porikli. Change detection by frequency decomposition: Wave-back. In *Proc. of Workshop on Image Analysis for Multimedia Interactive Services*, 2005.

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

[13] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, 37(3):297–336, December 1999.

[14] C. Spampinato and S. Palazzo. Enhancing object detection performance by integrating motion objectness and perceptual organization. In *To appear on Proc. of the 21st International Conference on Pattern Recognition (ICPR'12)*, 2012.

[15] C. Spampinato and S. Palazzo. Hidden markov models for detecting anomalous fish trajectories in underwater footage. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2012.

[16] C. Spampinato, S. Palazzo, and D. Giordano. Evaluation of tracking algorithm performance without ground-truth data. In *ICIP 2012*, pages 1345–1348, 2012.

[17] C. Spampinato, S. Palazzo, D. Giordano, Isaak Kavasidis, Fang-Pang Lin, and Yun-Te Lin. Covariance based fish tracking in real-life underwater environment. In *VISAPP (2)*, pages 409–414, 2012.

[18] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *J. Math. Imaging Vis.*, 18(1):17–33, January 2003.

[19] 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.

[20] Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, Yoichi Sato, and Yozo Fujino. Learning motion patterns and anomaly detection by Human trajectory analysis. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 498–503, 2007.

