

# A Video Processing and Data Retrieval Framework for Fish Population Monitoring

Emma Beauxis-Aussalet  
CWI  
Information Access Group  
Amsterdam, The Netherlands  
emma@cwi.nl

Simone Palazzo  
University of Catania  
Department of Informatics and  
Telecommunication  
Engineering  
Catania, Italy  
simone.palazzo@dieei.unict.it

Gayathri Nadarajan  
University of Edinburgh  
School of Informatics  
Edinburgh, United Kingdom  
gaya.n@ed.ac.uk

Elvira Arslanova  
CWI  
Information Access Group  
Amsterdam, The Netherlands  
elvira@cwi.nl

Concetto Spampinato  
University of Catania  
Department of Informatics and  
Telecommunication  
Engineering  
Catania, Italy  
cspampin@dieei.unict.it

Lynda Hardman  
CWI  
Information Access Group  
Amsterdam, The Netherlands  
lynda.hardman@cwi.nl

## ABSTRACT

In this work we present a framework for fish population monitoring through the analysis of underwater videos. We specifically focus on the user information needs, and on the dynamic data extraction and retrieval mechanisms that support them. Sophisticated though a software tool may be, it is ultimately important that its interface satisfies users' actual needs and that users can easily focus on the specific data of interest. In the case of fish population monitoring, marine biologists have to interact with a system which not only provides information from a biological point of view, but also offers instruments to let them guide the video processing task for both video and algorithm selection. This paper aims at describing the system's underlying video processing and workflow low-level details, and their connection to the user interface for on-demand data retrieval by biologists.

## Categories and Subject Descriptors

H.4.1 [Office Automation]: Workflow Management; H.5.2 [User Interfaces]: User-centered design; H.3.3 [Information Search and Retrieval]: Query formulation; I.2.10 [Vision and Scene Understanding]: Video analysis

## Keywords

Video Analysis; Data Visualization; Intelligent Workflow

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
MAED'13, October 22, 2013, Barcelona, Spain.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2401-4/13/10 ...\$15.00.

<http://dx.doi.org/10.1145/2509896.2509906>.

## 1. INTRODUCTION

The Fish4Knowledge project is continuously collecting video samples of coral reef fish, from 9 fixed underwater cameras, and since 3 years. The collection is too large to be analyzed manually, and biologists seek for a tool for automatic video analysis and for the retrieval of relevant video data. To address such needs, we explored the following research question: What data extraction and retrieval framework is suitable to process such large video collection?

This motivated the development of video analysis components that are dedicated to i) the detection of fish amongst other objects; ii) the tracking of single fish along their trajectory; and iii) the recognition of fish species. We also developed an advanced workflow management component that is able to dynamically handle video analysis processes over the large collection of video. The workflow engine can handle both on-demand user queries for specific video processing, and batch processes for the on-going analysis of the continuous collection of video samples. These components are linked to a user interface that provides interactive data visualizations and allow user queries for high-priority data analysis.

To study user requirements, we conducted two series of user interviews to collect the requirements for the video data retrieval system. The first series of interviews was conducted with 3 marine biology experts, specialized in coral reef fish. It allowed us to elicit the most important data to be retrieved from the videos. We then implemented several prototypes of the user interface, and conducted a second series of user interviews to refine the user information needs and the related user interface functionalities. The interviews and prototype refinements were conducted in an iterative process. A total of 34 researchers within the coral reef biology community in Taiwan and in the Netherlands participated in our study. We present here the components of our system for video data extraction and retrieval, and expose the mechanisms developed for dynamic and flexible data analysis.

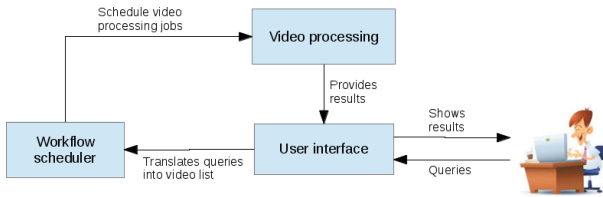


Figure 1: Framework overview.

## 2. RELATED WORKS

So far, research in underwater video processing has been mostly limited to constrained environments [4, 10, 14] such as tanks, where lighting conditions are controlled and stationary, the background is static, and the possible fish species are known in advance. These requirements do not hold in a open sea context, which greatly increases the difficulty for a automatic algorithms to process this kind of video data. However, underwater imaging has recently gained interest and technological improvements, as reviewed in [9].

Particularly relevant works are these of Spampinato *et al.*: [16] which represents one of the first attempts at detecting and counting fish from underwater videos, and [19] where a fish tracking algorithm is devised and tested on similar scenarios. Further, Kavasidis *et al.* [8] present a survey of motion detection algorithms performed with specific reference to fish in unconstrained environments. The evaluation of these algorithms is discussed in [18, 17].

Grid and Cloud workflow systems have emerged as fore-runners in providing a specialised environment to simplify the programming effort required by scientists to orchestrate a computational science experiment. Thus, Cloud-enabled systems must facilitate the composition of multiple resources, and provide mechanisms for creating and enacting these resources in a distributed manner. This requires means for composing and executing complex workflows, which has attracted considerable effort especially within the workflow community.

Existing workflow systems [3, 5] require the user to compose workflows, usually in a drag-and-drop or textual manner, or with some kind of aid of a verification tool. The Fish4Knowledge workflow system can automatically compose video processing workflows using domain knowledge and heuristics-based selection of steps in a planning-based manner [11]. In the absence of user-provided parameters for fish detection or tracking tasks, the workflow selects optimal tools using its planner. It also selects optimal resources from a set of heterogeneous distributed resources (with different resource schedulers), sends the workflow jobs to these resources, sets their execution priority and monitors the jobs for faults and exceptions.

## 3. VIDEO PROCESSING

The most basic level of data gathering in the system consists in processing underwater videos with computer vision algorithms for fish detection, tracking and species recognition. These modules are linked to the rest of the system as shown in Figure 1:

- The workflow modules receives user requests and translates them into calls to the video processing executable software;

- The results generated by the video processing are stored into a database which is used by the user interface querying module to extract meaningful information.

### 3.1 Fish detection

The object detection consists in identifying for each frame the location of all moving *objects of interest*. Typically it consists of: i) a *motion detection* phase: image pixels are individually analyzed and marked as part of the background of the scene (i.e. static elements) or of the foreground (i.e. dynamic/moving elements); and ii) a *blob extraction* phase, where contiguous regions of foreground pixels are joined into *blobs*, representing moving image regions, and potentially containing objects of interest.

Given the variety of scenarios and application scopes, no unique algorithm exists which is able to solve such problems. This is especially true in the underwater environment, since, as seen in Section 2, little scientific research has focused on this kind of environment. Moreover, underwater videos present complications not found in urban contexts. First, due to the technical communication difficulties between the underwater cameras and the storage servers, and due to the storage limitations the video quality is relatively low. Secondly, natural phenomena such as the gleaming of sunlight on the water surface and on the sea bed, introduce a kind of random rapidly-varying visual noise which may be detected as a moving object. Thirdly, other non-fish objects may be present in the observed scene (i.e. plants and algae).

We employed the ViBe [1] algorithm. According to [8] it outperforms state-of-the-art motion detection algorithms in the underwater environment. Unlike classical approaches, it does not presume the existence of an underlying statistical distribution describing the intensity values of a pixel in time. It handles a list of the most recent values, which is compared to the pixel’s current value to decide whether it is part of the background or the foreground. It also employs a random history updation policy, rather than a more common FIFO policy. This helps modeling quasi-periodic motion (typical of plants), characterized by the repetition of certain intensity values in time.

To reduce the influence of the phenomena causing the motion detection algorithm to mistakenly identify visual noise as foreground regions, we adopted a detection post-processing method to reduce the number of false positives [17]. Each detected blob is analyzed to extract a set of numeric properties describing its shape, color and motion. A Naive Bayes classifier is used to process this feature set and decide whether the blob is a fish or not.

### 3.2 Fish tracking

The fish identified in each frame are tracked across multiple frames. Detections from different frames are linked as being instances of the same fish. In a population monitoring framework this step is extremely important: it is the basis for a correct estimation of the population size. We deliberately used the term *estimation* because we can not evaluate the exact number of individual fish (e.g., fish can swim many times in and out the field of view).

The same difficulties described in the previous section also indirectly influence fish tracking, since the output of the object detection step is the input for the tracking algorithm. Other factors influence the accuracy of underwater tracking, such as the presence of several natural elements (such as

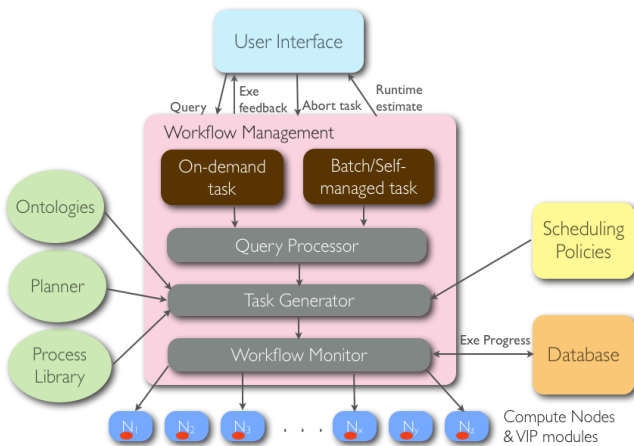
rocks, plants, corals) which may temporarily hide fish, or their motion properties. Unlike people-centred scenarios, fish may move in all directions, which may cause the fish size and appearance to quickly change. Such typical erratic motion pattern features rapid and sudden accelerations and direction changes, which complicates the tracking of fish.

To tackle these problems, we adopted an algorithm based on a covariance model of a fish appearance [15], which has been successfully tested on underwater videos [19]. The covariance model describes a set of pixel-based features (e.g., location, colour, intensity derivatives) as a covariance matrix describing the internal variability of these features. It provides a compact way to model both statistical and spatial properties of an object.

From the fish detection and tracking components, a set of 66 types of features are extracted. They describe the shape, color, texture, body parts and trajectory of fish. With these features, the species recognition component is able to recognize a set of 36 species. The species recognition algorithms are described in [6].

#### 4. JOB DISPATCHMENT AND EXECUTION

The workflow component of the F4K project is responsible for the composition, execution and monitoring of a set of Video and Image Processing (VIP) modules on High Performance Computing (HPC) machines. It can interpret user requirements (from the User Interface component) as high-level VIP tasks. It creates workflow jobs based on the procedural constraints of the modules (VIP components), and schedules and monitors their execution in a heterogeneous distributed environment (HPC component).



**Figure 2: 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.**

The workflow manager’s architecture diagram (Figure 2) shows an overview of the components that the workflow interacts with, its main functions, and its sub-components. Two types of queries are dealt by the workflow: on-demand queries from the user interface, and internal batch queries.

On-demand queries are those that originate from the user. They have high priorities and should be processed immediately. The on-demand queries as currently implemented in the user interface are listed below. The workflow is also able to process higher-level queries, such as “What is the overall fish population in the Nuclear Power Plant (NPP-3) station between January and March 2012?”. Such queries avoid users to directly interact with video analysis algorithms, which may involve technical choices beyond their expertise. High-level queries can be mapped into the low-level queries listed below.

- Q1 Detect and track all the fish in a given date range and set of camera locations.
- Q2 Identify all the fish species in a given date range and set of camera locations.
- Q3 Estimate how long a detection or recognition query will take to produce results, without sending the query for execution.
- Q4 Abort a query that is currently being processed.

Internal batch queries are those that are invoked by the workflow management system itself. This is predominantly batch tasks on newly captured video clips that involve fish detection and tracking (Q1 above) and fish species recognition (Q2 above). These batch queries are considered to have low priority and are scheduled to be executed at “quiet” times, i.e., when on-demand queries are least likely to be processed.

At present the computing environment consists of nine nodes on a cluster of virtual machines with a total of 66 CPUs (called VM cluster) and two nodes on a supercomputer with a total of 96 CPUs (called Windrider). The workflow system resides in the master node of the VM cluster and makes use of both platforms to process queries. It deals with two different resource schedulers: Gridengine (SGE) [13] on the VM cluster and Load Sharing Facility (LSF) [7] on Windrider. At present all high priority jobs, i.e., on-demand user queries for specific video processing, are scheduled on the Windrider platform. The low priority jobs, i.e., the internal batch queries, are split equally between the VM cluster and Windrider.

Batch queries are initiated internally by the workflow. At present, the workflow looks for new *unprocessed* videos over the last 24 hours from the database. If no new unprocessed videos are present, it looks for unprocessed historical videos and creates fish detection and tracking queries in the database which will be caught by the workflow engine.

After selecting the appropriate platform for a query, the workflow engine retrieves each video associated with that query. For each video, the command line call including the invocation of the resource scheduler and the selection of appropriate algorithms for that query type will be generated by the workflow engine. Such command line is known as a *job*. The generation of jobs is done via a planning-based workflow composition mechanism [11]. Using this mechanism, new software modules with enhanced algorithms can be easily added, detected and selected by the workflow engine.

Another important feature of the workflow engine is in dealing with job dependencies. This scenario applies to fish species recognition jobs (Q2). A fish recognition module can

only be applied to a video when a fish detection module has *already* processed it. The workflow engine deals with a fish species recognition job as follows:

- If fish detection has been completed on the video, then run fish recognition only.
- If fish detection has not been started, run fish detection and fish recognition, specifying a dependency flag between them.
- If fish detection has been started but not completed yet, run fish recognition with a dependency flag on the running fish detection job.

The workflow schedules a maximum of 300 batch jobs every 24 hours (i.e. we record 300 videos each day) and listens for new on-demand queries every 10 seconds. Implementation-wise, there are two daemon processes for i) managing the queries (every 10 seconds); and ii) creating batch jobs (every 24 hours).

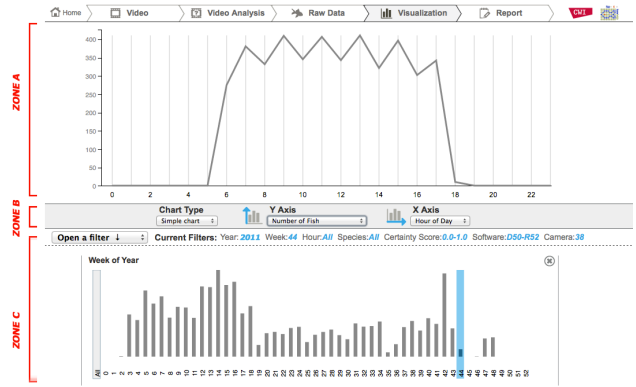
When the jobs are executing, the workflow monitors them for successful completion, or deals with errors that occur. The workflow can handle various scenarios on the Windrider facility, and development is on-going to have similar handling strategies on the VM cluster. They both use different resource schedulers and different mechanisms are needed to deal with these two schedulers. Factors influencing the performance of the workflow system, such as software capabilities and resource specifications, are contained in domain ontologies described in [12].

## 5. USER QUERIES AND DATA VISUALIZATION

The Fish4Knowledge system supports interactive visualizations of the data extracted from the videos. User requirements were studied by interviewing marine biology experts. The main user requirements concern the analysis of variations of numbers of fish over time and location. Further information needs, such as the analysis of migration or reproduction patterns, are not addressed since our video data does not currently offer a sufficient coverage of areas of interest. Other information needs, such as the analysis of food chain or cryptic species, are not addressed since video analysis can not supply the required information. More details are available in a project deliverable<sup>1</sup>. User needs for data provenance information were investigated in detail, since the acceptance of video analysis tools for scientific research relies on the ability of scientists to assess the provenance and accountability of the data. As reported in [2], biologists need extensive data provenance information, covering i) explanations of the video processing steps; ii) ROC evaluations; and iii) the sampling method.

This paper focuses on user needs for visualizing fish counts over time and location. Fig. 3 shows a visualization of the variations of fish counts over the hours of the day. The **Zone A** of the interface contains the main visualization that displays the fish counts. The **Zone C** contains *filtering widgets* that supports both: i) the selection of the dataset of interest (e.g., the timeframe, location and species of interest), for instance the timeframe in Fig. 3 is set to *Week 44* of *Year*

<sup>1</sup><http://homepages.inf.ed.ac.uk/rbf/Fish4Knowledge/DELIVERABLES/Del21.pdf>



**Figure 3: The User Interface that provides visualizations of fish counts.**

2011, and in *Zone A* the count at 8am is the sum of all fish appearing at 8am each day of that week; and ii) the overview of fish counts that can be obtained with different parameters (e.g., fish counts per location, per species, per year, etc). For instance, in Fig. 3, the *Zone C* displays the distribution of fish counts over weeks of the year, as obtained with the selected parameters (e.g., at *Camera 38*, of *Year 2011*). More filtering widgets can be opened on-demand (e.g., to select species of interest, or data from a specific versions of the video analysis software).

The **Zone B** of the interface supports the adaptation of the main visualization to specific user needs. Users can specify what is represented by the axes of the main visualization. For instance, while the y-axis represents fish counts, the x-axis can represent the distribution of such counts over either the weeks of the year, the hours of the day, or the locations. Additionally, users can select other type of graph (e.g., stacked chart, or box plot), which leads to the display of dedicated menu for adapting further the visualization. For instance fish counts can be stacked by species, as shown in Fig. 4. Such direct interaction with the axes of the graph was easy to understand for users, while offering a large choice of visualizations. This part of the interface allows a flexible data analysis. It can suit a wide range of user needs, in a context where biologists can seek for very different data analysis depending on their specific research goals.

Beside querying for fish count data, biologists can also query for data related to the number of video samples that are available, i.e., that were processed by the video analysis workflow. All video samples last 10 minutes, and can be processed by one or many versions of the software components. Video samples may be unprocessed, being in the workflow queue for their processing to be executed later. For instance, on-demand queries to workflow may consist of requesting the processing of the unprocessed videos needed for the current visualization. Besides, some videos may be discarded due to encoding errors. For instance, the Fig. 5 shows a visualization of the number of video samples from which the fish counts in Fig. 3 were extracted. They were processed by the versions *D50* of the Fish Detection & Tracking component, and the version *R52* of the Species Recognition component. To avoid confusion, the visualized data is always extracted

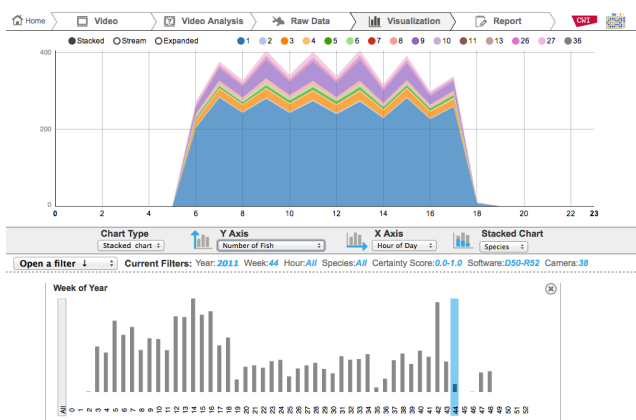


Figure 4: The User Interface that provides visualizations of fish counts stacked by species.

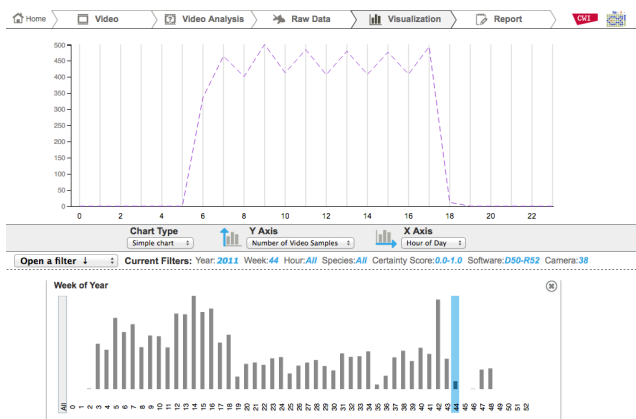


Figure 5: The User Interface that provides the number of video samples that are processed, and from which fish counts are extracted.

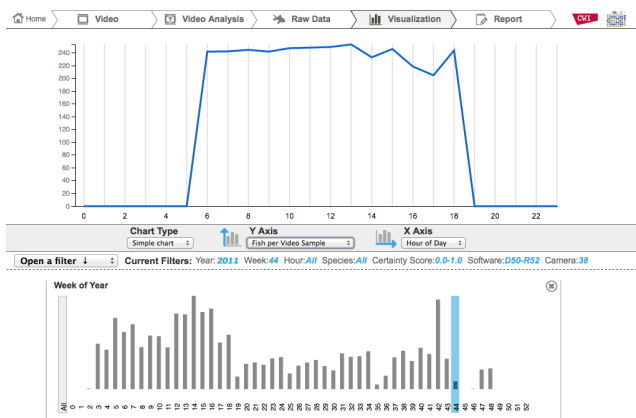


Figure 6: The User Interface that provides the average number of fish video sample, which are all 10 minute long.



Figure 7: The User Interface that supports user request for specific video analysis.

from one single version of each component, and never mixes data produced by several versions of the same component.

The variations in the numbers of video samples have a direct impact on the analysis of fish counts, e.g., the more video samples, the more fish. Our tool provides a simple mean to compensate for these variations. Biologists can visualize the average number of fish per video sample, as shown in Fig. 6. However potential biases still remain, since the risk of under- or over-representing fish is more important when the number of samples is smaller. Users require the number of samples to be homogeneous, so that potential under- or over-representations remain the same over the whole time periods and locations of study. This condition allows biologists to draw conclusions on the trends that can be observed in the fish counts. For this reason, our system supports user queries for on-demand video processing. Users can prioritize the analysis of specific sets of videos, and the workflow will reschedule the related jobs accordingly. This functionality is useful for cases were i) the number of video samples is uneven; ii) the number of video samples is too small; iii) users want to compare results obtained from different versions of the video analysis components.

Fig. 7 shows the user interface for querying on-demand video processing. Users can query the system for analyzing all available video samples for the periods and cameras for which they are currently visualizing fish counts. They can select the desired versions of the video analysis components, and ask the workflow to estimate the time needed to execute these components. When a query for video processing is launched, users can follow their execution on the right part of the screen, and potentially cancel the query if no longer relevant.

The user interviews we conducted allowed us to elicit user requirements for this data retrieval system and confirmed its usefulness. Future work on the user interface involves the implementation of advanced measures and visualizations of other potential biases in fish counts (e.g., over- or under-representation of specific species, impact of video quality), advanced features for multidimensional visualizations, and end-to-end usability test.



## 6. CONCLUSIONS

We described a system for data extraction and retrieval in a large video collection, and a user interface for interactive data visualization and user queries for specific video processing. The system includes video analysis components that use state-of-the-art video analysis algorithms for the detection, tracking, and species recognition of coral reef fish. We presented a workflow that manages these components, and that is able to handle both i) internal batch queries, continuously processing the video collection; and ii) on-demand user queries for specific video processing and the related rescheduling of the video processing jobs. Our user interface supports the retrieval of video data for fish population monitoring, with means for controlling and correcting potential biases in the retrieved data (i.e., the potential biases induced by uneven numbers of video samples). It provides flexible data visualizations so as to address a wide range of user needs, since biologists can have various information needs. Our system contributes to biology research for coral reef fish monitoring by addressing the user needs for data extraction and retrieval in large video collection, with a particular focus on the flexibility needed for both i) the dynamic scheduling of video processing tasks; and ii) the monitoring of specific aspects of fish population with adaptable data visualizations.

## 7. 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, E. Arslanova, L. Hardman, and J. van Ossenbruggen. A case study of trust issues in scientific video collections. In *Proceedings of the 2nd ACM international workshop on Multimedia analysis for ecological data*. ACM, 2013.
- [3] E. Deelman, D. Gannon, M. Shields, and I. Taylor. Workflows and e-Science: An Overview of Workflow System Features and Capabilities. *Future Generation Computer Systems*, 25(5):528–540, 2009.
- [4] F. H. Evans. Detecting fish in underwater video using the EM algorithm. In *Proceedings of the 2003 International Conference on Image Processing, 2003 - ICIIP 2003*, volume 3, pages III – 1029–32 vol.2, 2003.
- [5] K. Görlach, M. Sonntag, D. Karastoyanova, F. Leymann, and M. Reiter. Conventional workflow technology for scientific simulation. In *Guide to e-Science*, pages 323–352. Springer, 2011.
- [6] P. X. Huang, B. J. Boom, and R. B. Fisher. Underwater live fish recognition using a balance-guaranteed optimized tree. In *Computer Vision-ACCV 2012*, pages 422–433. Springer, 2013.
- [7] IBM. *Load Sharing Facility (LSF)*. <http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/products/lsf>.
- [8] I. Kavasidis and S. Palazzo. Quantitative performance analysis of object detection algorithms on underwater video footage. In *Proceedings of the 1st ACM International Workshop on Multimedia Analysis for Ecological Data, MAED12*, pages 57–60, 2012.
- [9] D. M. Kocak, F. R. Dalglish, F. M. Caimi, and Y. Y. Schechner. A focus on recent developments and trends in underwater imaging. *Marine Technology Society Journal*, 42(1):52, 2008.
- [10] E. F. Morais, M. F. M. Campos, F. L. C. Padua, and R. L. Carceroni. Particle Filter-Based Predictive Tracking for Robust Fish Counting. *Computer Graphics and Image Processing, Brazilian Symposium on*, 0:367–374, 2005.
- [11] G. Nadarajan, Y. H. Chen-Burger, and R. B. Fisher. Semantics and Planning Based Workflow Composition for Video Processing. *Journal of Grid Computing, Special Issue on Scientific Workflows*, 2013. (in press).
- [12] G. Nadarajan, C.-L. Yang, and Y. H. Chen-Burger. Multiple Ontologies Enhanced with Performance Capabilities to Define Interacting Domains within a Workflow Framework for Analysing Large Undersea Videos. In *International Conference on Knowledge Engineering and Ontology Development*, 2013. (to appear).
- [13] Oracle. *Open Grid Scheduler (SGE)*. <http://gridscheduler.sourceforge.net/>.
- [14] R. J. Petrell, X. Shi, R. K. Ward, A. Naiberg, and C. R. Savage. Determining fish size and swimming speed in cages and tanks using simple video techniques. *Aquacultural Engineering*, 16(63-84), 1997.
- [15] F. Porikli, O. Tuzel, and P. Meer. Covariance Tracking using Model Update Based on Lie Algebra. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.
- [16] C. Spampinato, Y. H. Chen-Burger, G. Nadarajan, and R. B. Fisher. Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos. In *3rd International Conference on Computer Vision Theory and Applications, VISAPP 2008*, pages 514–519, 2008.
- [17] C. Spampinato and S. Palazzo. Enhancing Object Detection Performance by Integrating Motion Objectness and Perceptual Organization. In *Proceedings of the 21st International Conference on Pattern Recognition, ICPR*, pages 3640–3643, 2012.
- [18] C. Spampinato and S. Palazzo. Evaluation of Tracking Algorithm Performance without Ground-Truth Data. In *IEEE International Conference on Image Processing, to appear*, 2012.
- [19] C. Spampinato, S. Palazzo, D. Giordano, F. P. Lin, and Y. T. Lin. Covariance-based fish tracking in real-life underwater environment. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2012.