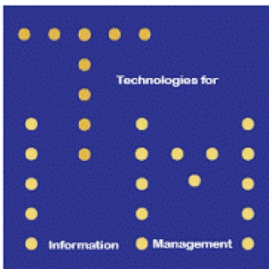
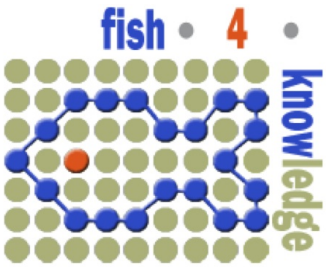




Fish4Knowledge Deliverable D2.3

Component-based prototypes and evaluation criteria

Principal Author:	E. Beauxis-Aussalet, J. He, C. Spampinato, B. Boom, J. van Ossenbruggen, L. Hardman
Contributors:	CWI, UCAT, UEDIN
Dissemination:	PU

Abstract: We provide an overview of the various user interface components developed and evaluated in the Fish4Knowledge project, and describe the development plans for future components and their evaluation criteria.

Deliverable due: Month 18

Executive Summary

This document describes an overview of the various user interface (UI) components developed and evaluated in the Fish4Knowledge project, and describes the development plans for future components and their evaluation criteria.

The work described here reflects an explicit change of focus when compared with the UI component development plans described in the Fish4Knowledge project proposal. This adaptation is based on two findings.

First, the need for ground truth for the training and evaluation of computer vision components within the project lead to the need for additional user interface support in various ground truth collection tasks, to an extent not foreseen in the original proposal. Section 2 describes the underlying research and user interfaces developed for collecting ground truth within the project, targeting both expert and lay users. It also discusses the quality of the ground truth data obtained with these user interfaces.

Second, user requirement studies identified a need to explicitly communicate uncertainty metrics and evaluation results to end users. In D2.1 *User Information Needs* [2], we sketched how the answers to almost all "20 questions" users might ask from the Fish4Knowledge system have associated issues to trust and uncertainty. To some extent, these issues can be regarded as general provenance questions that are relevant to all scientific data (including: where is this data coming from, when was it collected, who was responsible and what assumptions were made during data collection). More specific trust issues, however, are directly related to the inherent uncertainty introduced by the Fish4Knowledge computer vision components.

The need to be able to trace back the overall data provenance and to explicitly handle the uncertainty introduced by the computer-vision components was reflected in the *Charles scenario* described in Section 2.1 of D2.2 *User Scenarios and Implementation Plan'* [1]. Section 3 identifies 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. Section 4 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. In section 5 we discuss the evaluation criteria for future user development criteria and section. Section 6 concludes the document and outlooks to the future work.

Contents

1 Introduction

The goal of this section is to introduce the specific nature of the Fish4Knowledge user interface components, and why dealing with uncertainty in the user interface is so important, and what types of uncertainty we need to address.

In most computer applications, the correctness of the data is implicitly assumed. The task of the back end is to compute this data as efficiently as possible, and with the term *performance* people generally refer to the (CPU) time and (memory) space needed to do this computation. The task of the front-end is to give the user access to view and manipulate the data. If data turns out to be incorrect, this is considered a bug, something that needs to be corrected somewhere in the input data or the processing software.

In contrast, the key data in the Fish4Knowledge system, that is, the data about the raw video footage, is produced by computer-vision and other machine learning components that have an inherent margin of error. In this context, the term *performance* is typically related to the numbers of errors a component makes when doing its computations. Errors in the data are unavoidable, and, even if individual errors could be corrected manually, this if often not effective given the sheer size of the data set. As a result, errors in the data cannot be solved by fixing the input data or the processing software, and the errors will still be in the data communicated to the user.

Since the system is to be used in a scientific context, it is important for the user (e.g. the researcher) to know to what extent the error margins in the data obtained from the system are acceptable in the given context. For example, a certain observation in the data might reflect a real event in the fish population under study, but it might also be caused by inherently noisy data. The user needs to be provided with sufficient information to decide which of the two cases is more likely.

To support the user in this task, we first need to know what the error rates are for the selected data set. Second, we need to effectively communicate these rates to the user. The research challenges associated with these two problems and their implications on the design of the user interface are the main focus of this deliverable.

There are many different components in the Fish4Knowledge system that produce data, and each component has its own error characteristics. Typically, these depend on the exact parameters that were used to configure the component. Over time, components may evolve, resulting in other error characteristics. For each data entry, we thus need to record by which version and configuration of a particular component it was generated. We then need to evaluate all versions of each component used to determine their error characteristics. To be able to do this component-based evaluation, we need a ground truth for a representative and sufficiently large subset of the entire data set for each feature being detected. Given that such ground truth data does not exist for our domain, we thus need to create a number of ground truth data sets within the project.

Ground truth data is used for both training and evaluating components. Initially, components require training data to automatically learn how to classify their data, so for these components ground truth data is important in the training phase, that is, even before any real data has been produced. Ground truth data is also required in the component-based evaluation phase to measure the performance of a single component, and to compare the performance of different (versions of) components during development. While these are standard procedures, within Fish4Knowledge, ground truth data is also needed to communicate the error rates of trained and

evaluated components to the end user.

Collecting the necessary ground truth has been a challenging task, to which all research partners in the project have devoted considerable time and effort. Tool support turned out to be crucial in order to obtain ground truth data in sufficient quality and quantity, and different features require different tools and user interface designs. An additional challenge has been to reduce the time required of the experts within the project and to develop alternative interfaces that allow lay users contribute to the creation of the ground truth data.

1.1 Uncertainty in computer vision components

The key components of the Fish4Knowledge system include five computer vision components: fish detection, tracking, description, clustering and recognition (see Deliverable 5.1). For each of the components, their outputs are not error free. These errors can be captured using standard evaluation metrics as described in Deliverable 5.3. Each component generates a certainty score indicating the confidence in its output. The certainty score depends on the specific version of a specific component. The confidence score needs to be calibrated using a ground truth data set. Even after the certainty scores have been calibrated with the ground truth data, this still does not guarantee that using the same component on a different data set will actually reflect the "true" error detection rates.

The fish detection components of the system are used by other, higher-level, components in the system, so that the associated certainty scores will propagate throughout the system. For example, the certainty score in the fish detection components will propagate to the species recognition components.

1.2 Uncertainty in ground truth

While the uncertainty introduced by system errors can be measured using a number of evaluation metrics, the ground truth used for the evaluation introduces its own uncertainty. Within the project, the ground truth data for fish detection and tracking can be done relatively easily since its correctness can be verified without expert knowledge. The creation of ground truth for species recognition is, however, more difficult since i) non-experts make mistakes while identifying fish species, and ii) experts sometimes cannot reach an agreement on the species of a fish, see Section 2.2.1.

1.3 Communicating uncertainty to the user

The goal of the user interfaces within the project is to allow marine biologists to select sets of automatically analysed data that allow them to draw scientifically significant conclusions, (represented by the "20 questions" detailed in Deliverable 2.1). In order to achieve this goal, the user interface (UI) needs to allow expert users to understand where in the system uncertainty is introduced, and its likely effect on the conclusions that they wish to draw.

To support users in understanding and evaluating the uncertainty inherent to image processing, the user interface integrates specific uncertainty metrics, such as those described in Deliverable 5.3 (e.g., detection rate, false alarm rate).

Thus for each task, users are assumed to need two types of metrics: one that expresses a measure for the biological phenomenon of interest, and one that expresses a measure of

uncertainty about the first metric:

- **Population metrics**, such as those described in Deliverable 2.1. (e.g., counts of fish, growth rate, species composition), that describe the population dynamics of fish from specific species, time period, location or behavior.
- **Uncertainty metrics** that describe the types of errors that the automatically analyzed data is likely to contain.

While the uncertainty metrics described in Deliverable 5.3 are well understood in the computer vision community, they are not necessarily understood and accepted by users. In particular, the certainty scores produced by individual components need to be communicated to and understood by users. When components are used in other parts of the system, the influence of a certainty score on another component making use of it also needs to be conveyed. Ideally, a user should be able to carry out a high-level task with confidence that the system will produce only reliable results, and with the knowledge that questions about the certainty of the data analyses can be checked through interactions with the system.

2 UIs for component-based ground truth collection

Ground truth generation for the detection, tracking and recognition algorithms training and evaluation, as aforementioned, is inarguably the most time consuming and onerous task in the whole evaluation process stack, including evaluation over individual computer vision component as well as overall system output. Moreover, the accuracy of obtained evaluation results is directly proportional to the quality of the supplied ground truth. Given the importance of high quality ground truth generation, a tool for not only drawing efficiently accurate annotations, but also for combining multiple annotations in order to increase their overall quality, had to be developed.

In the following subsections, we introduce the methods and tool used for creating fish detection and tracking ground truth in Section 2.1, and that for creating fish recognition ground truth in Section 2.2.

2.1 Annotation tool for detection and tracking

Perla (PERformance evaluation, Labeling and Annotation)[1] is a client-server rich internet application which features a collaborative environment that allows users to share their own annotations with others. By increasing the number of annotations per video and integrating annotations from multiple users, it accelerates the high quality video ground truth generation process.

Perla offers a personal workspace where the user can find information about past activities. Through her workspace a user can also create, modify, review and share ground truth with other users, in order to implement a collaborative environment for large-scale video annotation acquisition (Fig. 1). Moreover, the tool offers the methods necessary to navigate through the plethora of processed videos of the project's image processing applications in an intuitive and easy way by using the integrated search engine (Fig. 3, right).In particular, it allows users

[1] http://f4k.ing.unict.it/perla.dev

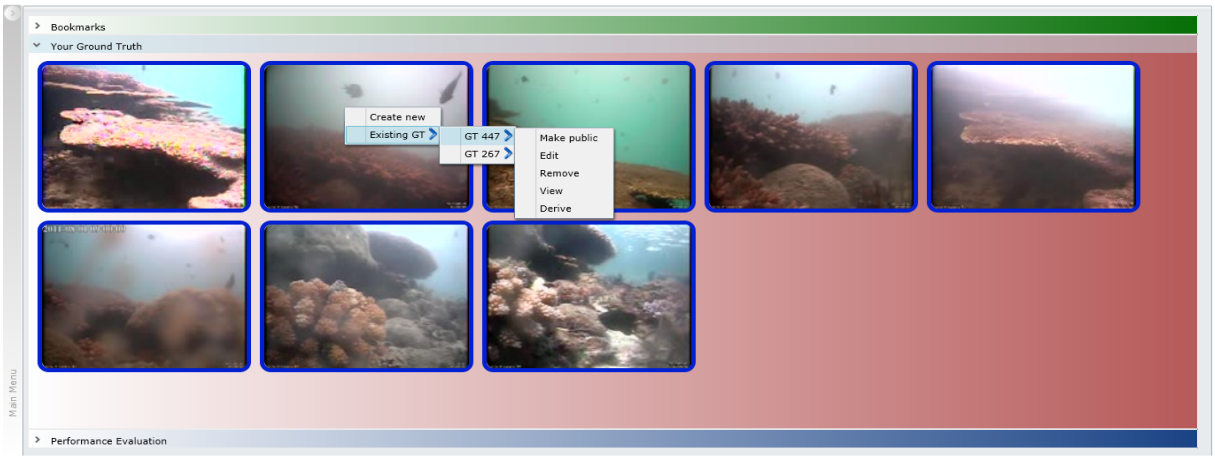


Figure 1: The workspace: ground truth management.

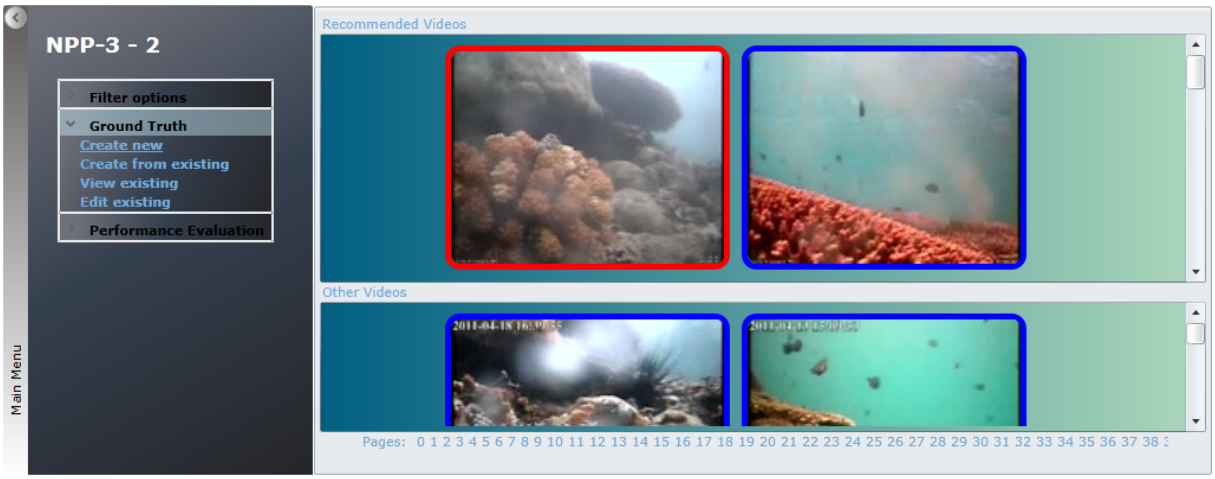


Figure 2: The video selection window

to limit the number of the shown videos by defining criteria regarding the videos' resolution, acquisition time, enabling the user to select videos with specific features(e.g. day or night). Once the user locates a video she is interested in, she can always bookmark it for future reference without having to pick it from the whole video collection.

2.1.1 Establishing ground truth for fish detection

Once the user identifies the videos she wants to create ground truth for, she can create annotations by using the provided multiple window application. Each drawing window (Fig. 3, top left) shows one image and, by using the available tools (Fig. 3, bottom left), annotations can be drawn on it. While the tools found on the toolbar are commonly found in many other ground truth generation applications, there are situations when these tools results are inefficient at best. For example, one of the most populated annotated videos in the project's repository, contained about 18000 fish. If the annotations were done by only using the manual tools and considering that annotating a single object needs on average about 15 seconds, the total time needed to annotate one densely populated 10 minute, low resolution, 5 fps videoclip would be about 75

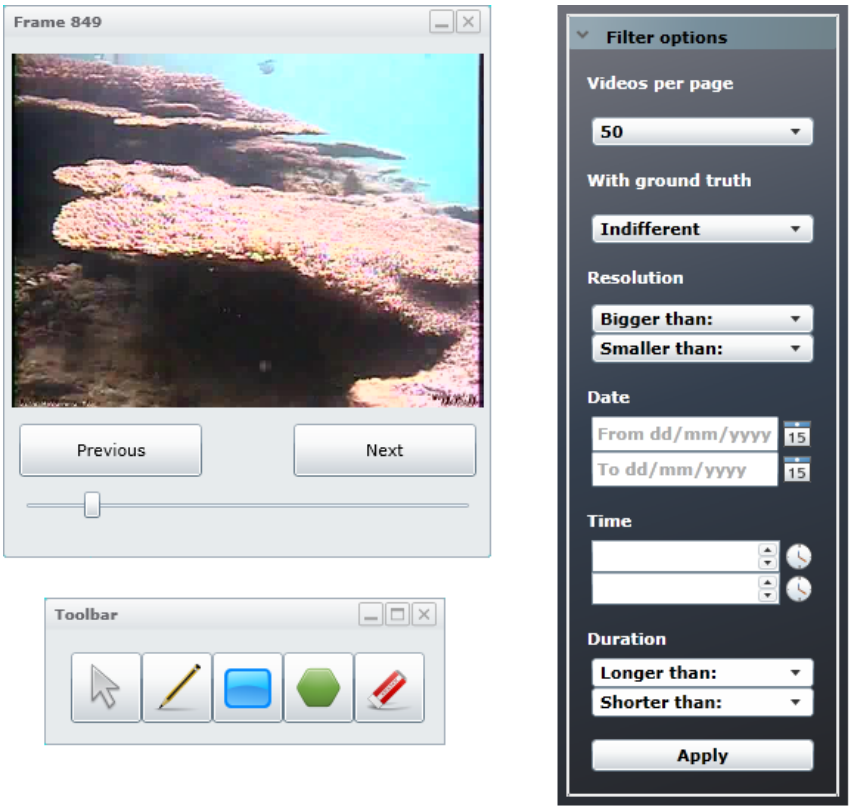


Figure 3: *Top left*: A drawing window that shows an image to be annotated. The controls located at the bottom of the window allow the user to navigate through the image sequence. *Bottom left*: The toolbar. From left to right, the Bounding Box Selection, Pencil, Rectangle, Polygon and Eraser tools. *Right*: The search engine provided.

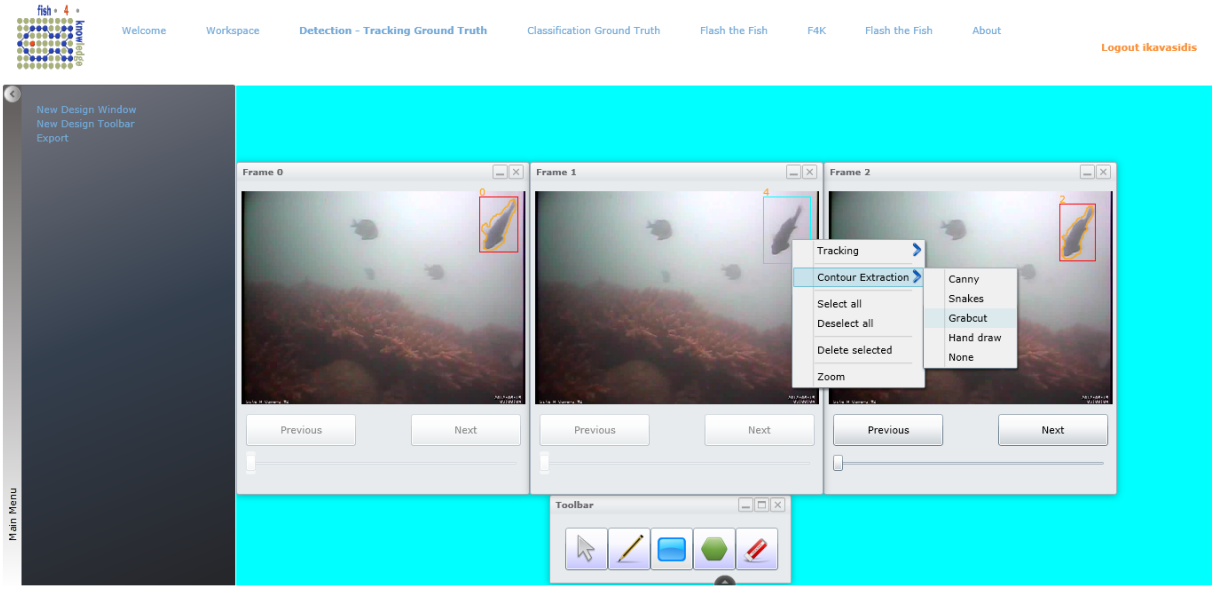


Figure 4: Semi-automatic contour extraction applied on the central drawing window.

hours! Under these conditions, assisting users in annotating objects as efficiently as possible seems necessary. To this end, Perla, offers three automatic contour extraction tools, namely, Grabcut, Snakes and Canny edge detection, in order to hasten the ground truth generation process (Fig. 4), especially in high contrast videos where these contour extraction algorithms work best.

2.1.2 Establishing ground truth for fish tracking

We now proceed to introducing the procedure of creating tracking groundtruth using Perla.

In Perla, the tracking ground truth generation exploits the advantages of multiple windows interfaces in order to provide an easy-to-use and intuitive way to follow objects across consecutive frames. In particular, by arranging two design windows side-by-side the user creates "drawing chains" (as the one in Fig. 4). While in a "drawing chain", the Next and Previous buttons and the slider of each drawing window in the chain are disabled except from the last one's (the rightmost), which serves as a control to navigate through the image sequence. When used in high resolution or multi-monitor desktop setups, the application can host multiple drawing chains enabling the user to annotate different parts of a video at the same time.

2.1.3 Combining multiple annotations

Every user is a different one. The annotations provided sometimes vary minimally but sometimes are substantially different among different users. While a well-done annotation (e.g. made by a graphic designer) could supposedly constitute the gold standard in ground truth generation, it is not always possible to acquire such an accurate annotation of the objects.

The web nature of Perla permitted us not only to implement a multi-user platform that enables collaborative video annotation but also to integrate methods that combine multiple annotations of the same object from different users in order to derive one single instance. This result is practically the combination of the best features of each of the annotations. The

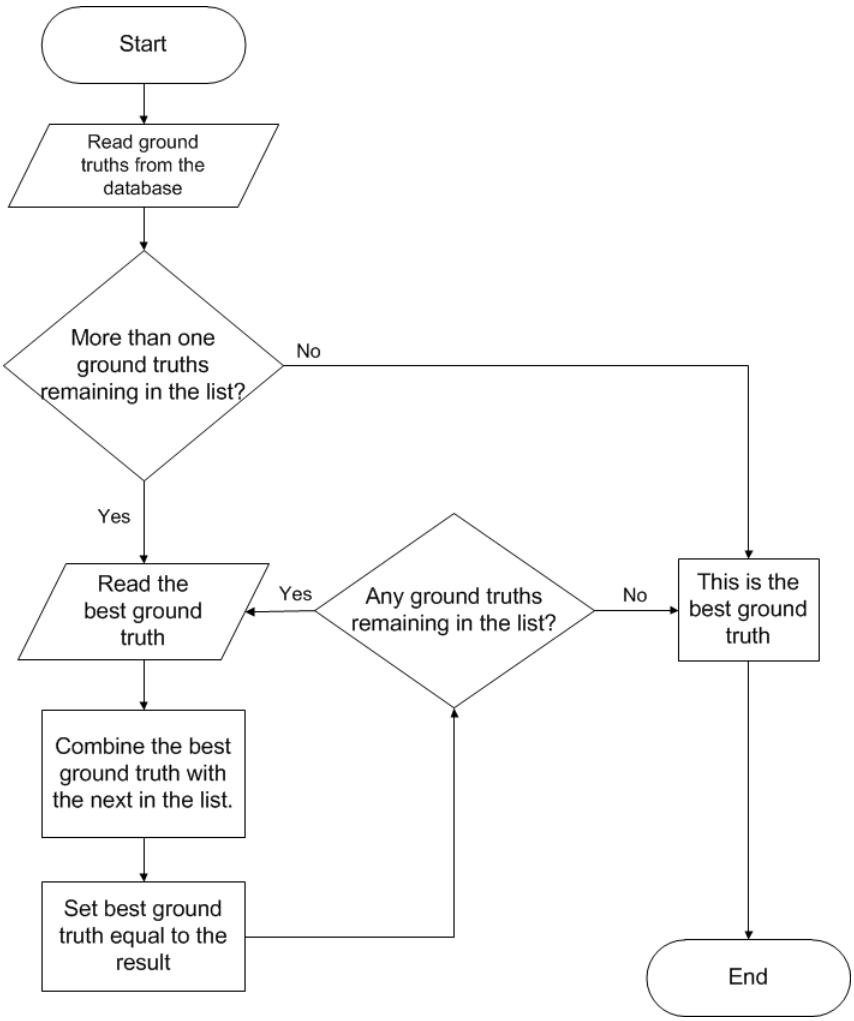


Figure 5: Flowchart of the "best ground truth" building process.

collaborative aspect of the platform aims at assisting the users in ground truth generation by allowing them to share their annotations. In particular, a user, instead of having to create a ground truth from scratch for a video for which annotations already exist, she can derive them and modify them at will. Alternatively, user groups can edit different parts of the same video at the same time, reducing substantially the amount of time needed.

Multiple annotations of the same object can be combined by employing a voting policy, in order to create better representations of the respective objects (the one herein called "best ground truth"). Building a "best ground truth" (BGT) involves two basic steps: i) adding new annotated objects to the BGT, ii) integrating contours (Fig. 5).

Supposing that the BGT has been already built for a given video and new annotations for the same video are created, then, for each new annotated object A, two cases may occur:

- **New object instance**. The object A did not previously exist and it is inserted to the BGT as is. This exploratory strategy avoids limiting the number of objects on each ground truth; however, to prevent noisy ground truths, each object instance in the BGT considers the number of annotators that have labeled it over the total number of annotators, thus

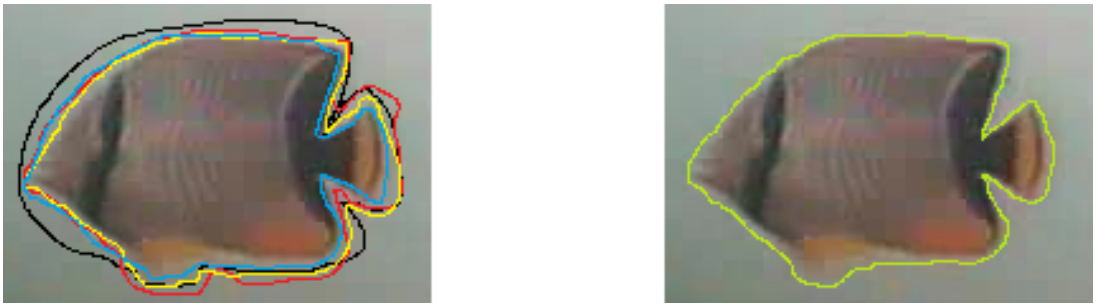


Figure 6: Building a "best ground truth" object. On the left, four annotations (black, yellow, red from different users and blue is the existing BGT representation). On the right, the resulting best ground truth.

allowing us to filter out object instances which were infrequently annotated.

- **Existing object instance**, i.e. there exists an instance (referred in the following as GT) of object A in the BGT. In this case, we assess a matching score between object A and object GT and if this score is greater than a given threshold (in our case 0.75) the contours of A will be combined with the ones of GT. A resampling of the object's contours usually is applied in order to equate the number of the points of the object A and the object GT. The matching score is the weighted mean of the two following measures:
  - Overlap Score. Given the object A and the corresponding object GT of the best ground truth BGT, the overlap score, O_{score}, is given by:

$$O_{score} = \frac{area(A \cap GT)}{area(A \cup GT)} \tag{1}$$

  - Euclidean Distance Score. Pairwise euclidean distance between A points (X, Y), with $(X_i, Y_i) \in A$, and GT points (x,y), with $(x_{i'}, y_{i'}) \in GT$, computed as:

$$E_{score} = 1 - \frac{\sum_i^n \sqrt{(X_i - x_{i'})^2 + (Y_i - y_{i'})^2}}{max(\sum_i^n \sqrt{(X_i - x_{i'})^2 + (Y_i - y_{i'})^2})} \tag{2}$$

Once a new object is considered for being part of the "best ground truth" (see above) its contours C_A are combined with the contours C_{GT} of the corresponding "best ground truth" object to form the new object contours C_{NGT}, where each point is computed as:

$$C_{NGT}(i,j) = \frac{1}{2^{N-1}} \sum_{n=1}^{N} (w_A \times C_A(i,j) + C_{GT}(i,j)) \tag{3}$$

where $w_A \in [T, 1]$ (where T is the threshold described above, and is set to 0.75) is the matching score between A and GT computed as above described and N is the number of different annotations for that given object. Fig. 6 shows the result of a combination of three annotations and the existing GT on the same object. Fig. 7, instead, shows how contours definition becomes more precise as the number of annotators increases.

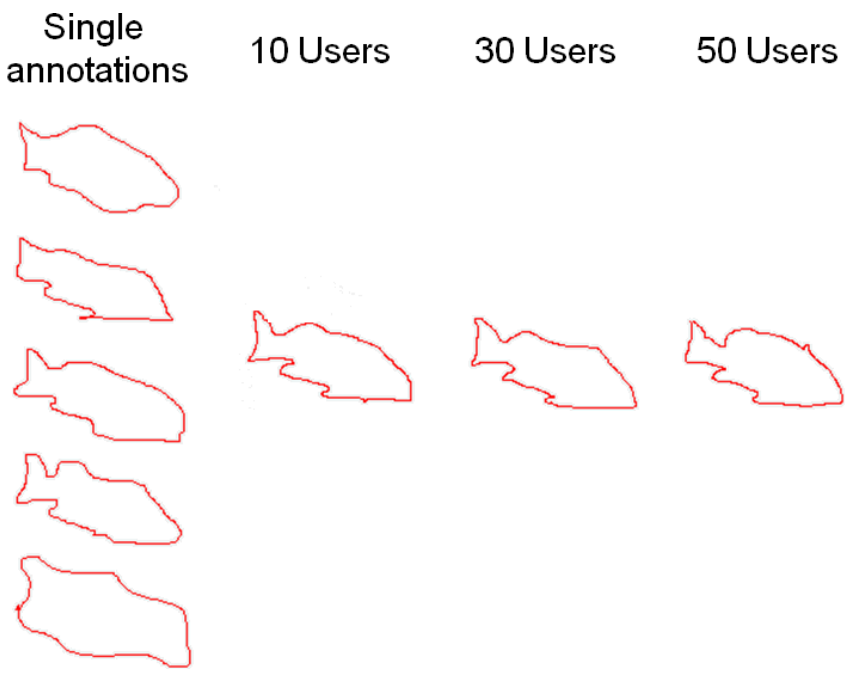


Figure 7: Object contours quality improves as the number of annotations increases

2.2 Annotation tools for fish recognition

In this section, we introduce the methods and tools used for creating ground truth for fish recognition.

In order to be able to train and evaluate fish recognition algorithms, we need to obtain ground truth about which fish images belong to the same species, along with the species names. To support the manual labelling of images, we use a cluster-based method to group and retrieve similar images, which allows us to label a large dataset in an efficient manner. We conducted a two-stage annotation procedure. In the first stage, we use a manual cluster-based approach to assist the expert annotators, i.e., the marine biologists to label a small subset of the available fish images. In the second stage, we use an automatic clustering based approach to support non-expert annotators to conduct large scale annotation.

2.2.1 Cluster-based interface for expert annotators

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 labeling 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 manually clustered 3678 images randomly chosen from our video data. We then present them in a labeling interface as shown in Figure 8. Using this interface, the expert annotator first enters the species name that applies to the majority of the images in a cluster in the top-right text box. Once the name is entered, all images within the cluster are automatically assigned with the same species name. Then, the annotator is asked to select those images that do not belong to the cluster. By selecting these images, he/she can input the correct species names for

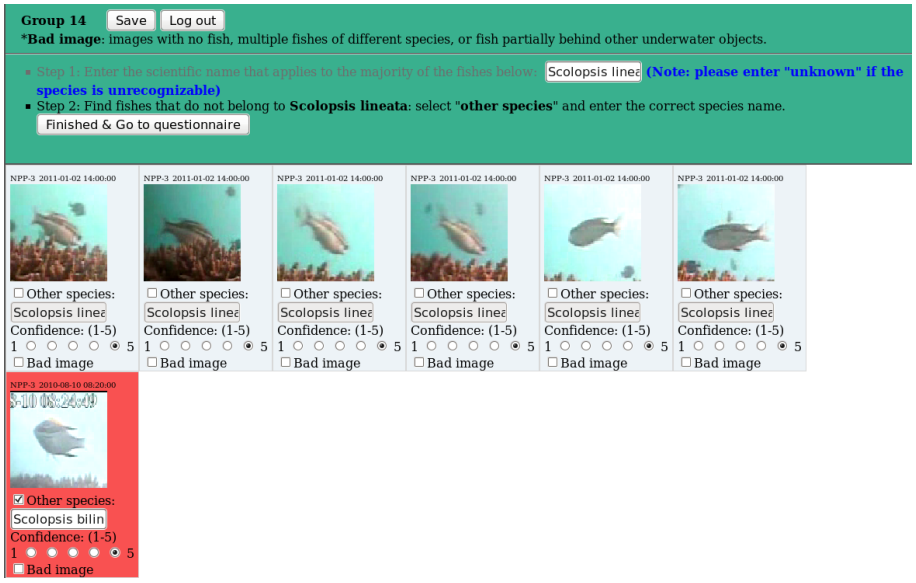


Figure 8: Interface for experts.

them in the text box under each image. In this manner, in the worst case, the annotator will have to manually assign a species name to each of the images, i.e., when the clustering is so bad that each image within a cluster represents a different fish species. In the best case, i.e., when the cluster is pure, the annotator only needs to enter the species name once. After finishing annotation, we also include a questionnaire for the experts in order to collect information such as whether the labeling task is difficult for him/her, and why it is difficult. To limit the amount of effort experts need to check the clusters, at most 30 images are randomly selected from each cluster and shown to the experts.

We invited 3 marine biologists (referred to as E1, E2 and E3) to participate the expert labeling task. They have research experience over 30, 10 and 25 years in the Taiwan sea area, respectively.

In total 190 images are labeled by the biologists. We notice that the marine biologists do not always agree on the species names for a given image. We use Cohen's kappa to measure the agreement between the expert labels, assuming the complete category set consists of all the species mentioned in the labels. See Table 1.

In addition, we notice that sometimes the biologists are not sure which species a fish should belong to, and they assign labels such as "A or B", or simply assigns a family or higher level label instead of a species level label. In the former case, we consider both labels mentioned, and in the latter case, we consider all species under a higher level label as possible target labels. Thus it is possible that an image has multiple labels assigned by a single expert. In total, 288 species were mentioned as labels for the 190 images. Since Cohen's kappa does not handle multiple labels of a single rater, we handle this situation as follows. First, we evaluate the agreement between labels at both species and family levels: it is expected that at family level, cases with such situation will be greatly reduced. Second, when there exist multiple labels for an image assigned by one expert, we randomly draw one of the them as the target label being evaluated; this process is repeated 100 times and we report the averaged κ and its standard deviation over the 100 runs. Note that the agreement calculated in this way is rather conservative.

Results in Table 1 show that at species level, the agreement between the experts are rather

	Species level		Family level	
Comparison	Avg.κ	Sdv.	Avg.κ	Stv.
E1 vs. E2	0.55	0.008	0.85	0.004
E1 vs. E3	0.48	0.008	0.75	0.000
E2 vs. E3	0.67	0.006	0.76	0.0001

Table 1: Cohen's kappa for measuring expert annotation agreement.

moderate, while at family level, a much stronger agreement can be found, but still not perfect. This result suggests that our labeling task is not trivial even for experts. Further, from the questionnaire we learn that according to the experts, the top factors that make recognition difficult are: 1) the low quality of the images and 2) the fact that some species are visually very similar.

2.2.2 Cluster-based large scale annotation

Although not able to name fish with their species names, non-experts have shown to be able to identify similar fish, e.g., in the previous experiment, only 6 out of 27 clusters contain wrongly clustered images. Hence we use non-experts to perform a cluster-validation task for a large scale species annotation. That is, instead of giving a label for every fish, we ask non-expert annotators to judge the quality of (automatically created) clusters.

We first create clusters using Affinity Propagation [3]. We choose Affinity Propagation as the clustering algorithm because it does not only cluster the images, but also selects a representative image for the cluster. We use this image to merge the clusters when the dataset is "over-clustered".

Our labeling method consists of three steps:

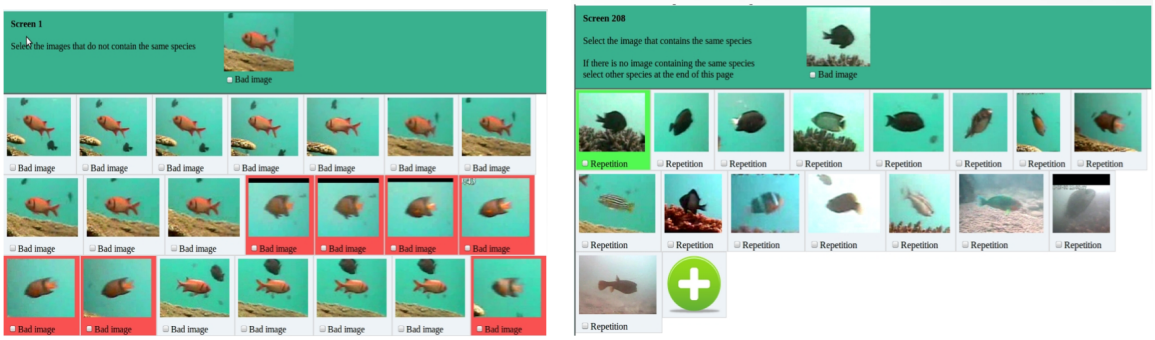
1. Cleaning the cluster, where we remove images which are not similar to the representative image;

2. Merging the clusters by linking the representative image of the cleaned clusters;

3. Linking removed images from the cleaning stage to the cleaned clusters.

After the three steps, we can assume that a cluster includes all fish of a certain species in the dataset. Whether one needs to perform the last step depends on whether all images in a dataset need to be labeled, or that a large subset of all images is sufficient. Figure 9(a) and 9(b) show the two non-expert cluster-validation interfaces. We use interface I for cleaning clusters, e.g., step 1, and interfact II for merging the clusters or merging the singleton images to the cleaned clusters, i.e., step 2 and 3.

Using the above described interface, we have a first dataset of 3678 fish images labeled. We found 6 annotators to annotate the entire dataset. Based on the labeling of the biologists, we found out that the average user performance achieves 87.6% correctly labelled fish. There is however a large difference between people who saw the fish images for the first time and people who are part of this project having observed some of the fish before. The lowest user performance is 68.8%, where this person basically annotated different species that look similar to the same category. For users, it is often very difficult to determine if fish belong to a different species or not, because appearances of the fish such as colour can change due to illumination



(a) Interface I (b) Interface II

Figure 9: Interfaces for non-experts.

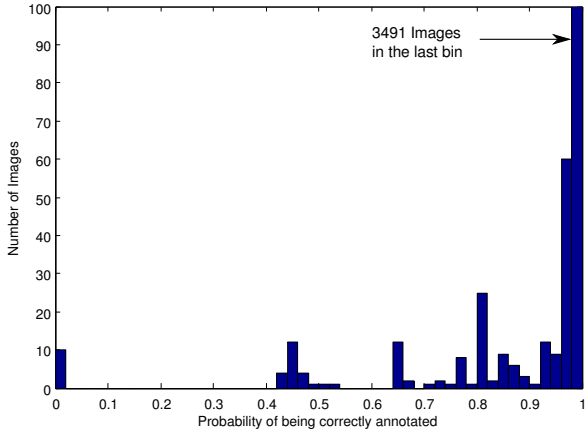


Figure 10: We show the distribution on the probabilities that an image labelled by multiple annotators is correctl. However, in this database, there are still a lot of disagreements between annotators. These images can be used to communicate to the marine biologists, for now they are excluded from the training and evaluation of recognition methods.

conditions. Further, when judging images with multiple fish or fish that they feel difficult to identify, some users used the ignore options very frequently, while other users used it rarely.

We estimate the probability that an annotator has correctly labeled (clustered) a fish using the expert annotations and then use it to combine the labels from multiple annotators. If annotators agree on a label, the probability becomes very high while with disagreements, the probability is much lower. In Figure 10, we show the distributions on the user's disagreements, in most cases however users do agree which can be observed in the last bar. In more than 90% of the images, the probability of being correctly labelled is greater than 99.9%.

This probability distribution can be seen as an estimation of the quality of the user performance in cluster-validation. Note that to correctly interpret the results stated above, we need to know: i) the species distribution among the labeled images, since some species are dominant and easier to identify than others; ii) whether the images in this datasets contain many continuous frames of the same fish, as it is easier to cluster these images compared to images of different fish of the same species in different environment.

Fish Species	Solitary	Pairing
Dascyllus Reticulatus	Abnormal	Breeding
Chromis Margaritifer	Normal	Breeding
Plectrogly-Phidodon dickii	*unknown*	Breeding
Acanthurus Nigrofuscus	Abnormal	*unknown*
Pomacentrus Moluccenis	Abnormal	Breeding
Chaetodon Trifascialis	Normal	Normal Breeding
Zebrasoma Scopas	Juvenile	Rare
Scolopsis Bilineate	Juvenile	Adult
Amphiprion Clarkii	*unknown*	Breeding
Siganus Fuscescens	Abnormal	*unknown*

Table 2: Interpretation of Solitary and Pairing Events depending on Fish Species

2.3 Fish behavior annotation

Based on the user studies specified in Deliverable 2.1, we understand that end-users are interested in fish behaviours related to demographics, reproduction, feeding, and environmental conditions. In order for the system to be able to identify this type of behaviours, we created an UI dedicated to the collection of corresponding training data. We focus on the 10 species whose detection, tracking and recognition results are available in the F4K database. We derived the specific fish behaviours of interest on the basis of descriptions of the 10 species provided by end-users and by the FishBase [2]. Here we investigate only pairing and solitary behaviours, as we assume they can be labelled by non-experts. The Table 2 summarizes the interpretations of fish co-occurrences. We report the following observations:

- Fish pairs, and solitary fish can contribute to the study of demographics and fish reproduction.
- The meaning of fish pairs, and solitary fish depend on the species involved.

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

The user interface functionalities support i) the retrieval of video excerpts that display the co-occurrences of interest, and ii) the manual selection of video excerpts that are suitable for the training dataset. It organizes the dataset collection task in 3 steps:

1. Define the rule, and the sampling method.
   Users are supported with 2 simple rules, and a set of parameters they can modify. The most important rule supports the retrieval of solitary fish and pairing fish. It covers most of the events of interest from Table 2 . Figure 11 shows how our user interface supports the specification of rule parameters.

[2] http://fishbase.org

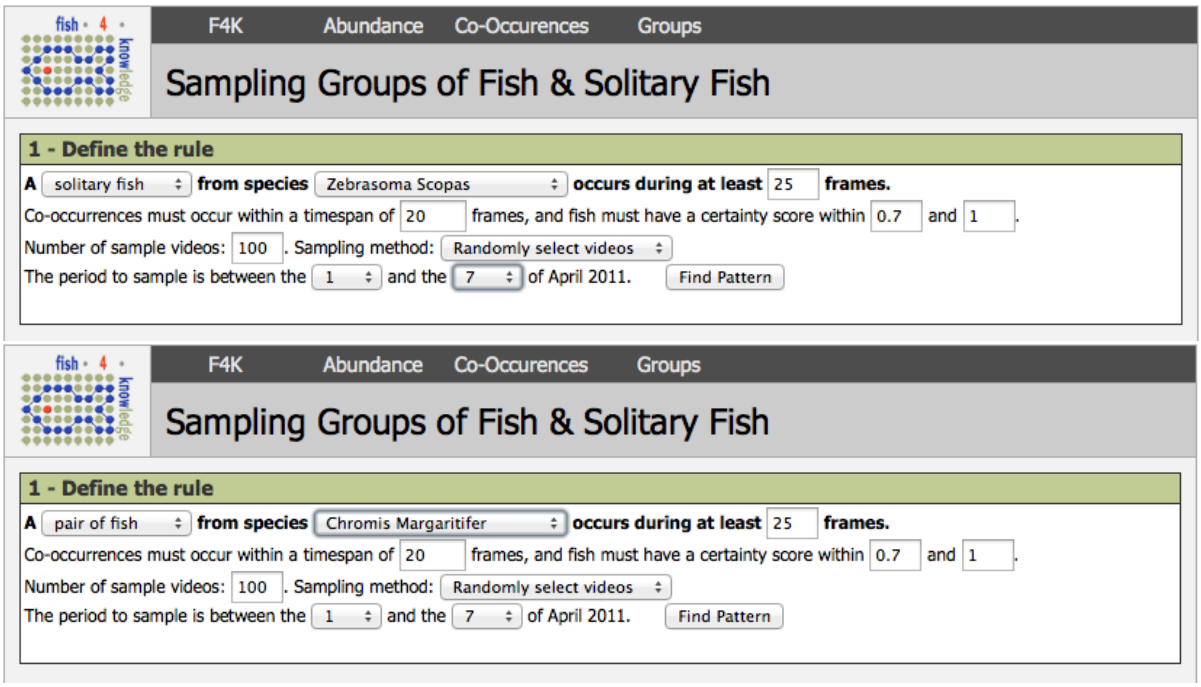


Figure 11: Screenshots of user-defined rules for retrieving solitary and paring fish.

2. Manually select valid video samples.
   Users are supported with a list of video samples that satisfy the rule they defined. Our system retrieves the video excerpts that display the co-occurrences of interest, as defined by the rule. Users can watch the video samples page by page. If a sample is a good example of the event of interest, users can click on the sample to include it in the training dataset. Fig. 12 shows a selected and a discarded video sample in our user interface.

3. Store the training dataset.
   After selecting a set of training video samples, users can label the training dataset and describe what event detection it supports. Fig. 13 gives an example of a label for a training dataset. When storing the dataset, the system saves the rule parameters and all the video samples it retrieved: the manually selected samples, flagged as valid samples, and the discarded samples.

3 Uncertainty and its impact on UI

The interfaces provided need to allow the user to identify selections of data that represent real-world effects, such as an increase in fish abundance. The data that is returned in response to a user query is not the end of the user's task, but the beginning of a process that allows the user to verify the validity of the real-world effect. Uncertainty in the ground truth data is inherent within the system, but its effects can be conveyed to the user to at least some extent. Uncertainty in the computer vision components is, however, likely to be the main concern of the experts and is the main topic we address in this section. We describe in detail how the uncertainty, in the ground truth data sets and the components created using them, affects how the results should

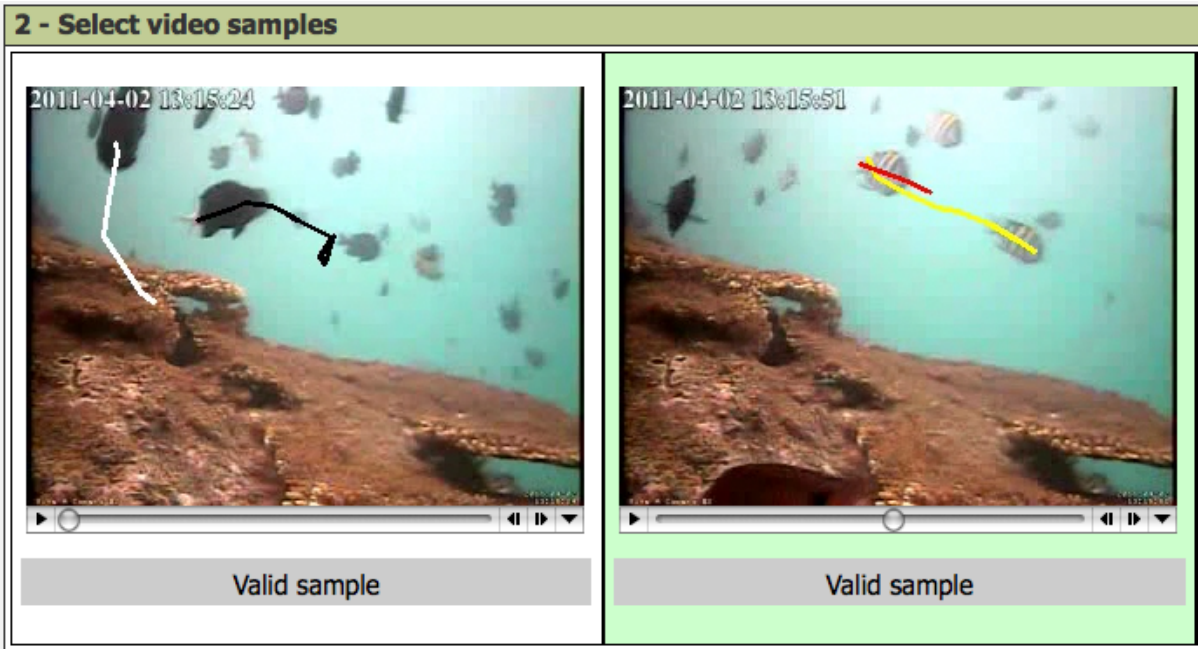


Figure 12: Users can select valid video samples (e.g., the video on the right is selected) and discard the others.

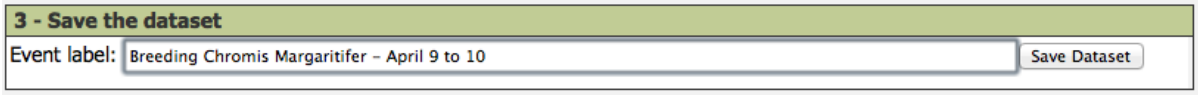


Figure 13: Users can label the training dataset to describe the targeted event.

be interpreted by the user. Within these descriptions we identify and summarise requirements for the user interface. The UI requirements stated here are based on general design principles and guidelines and the goals of the users stated in previous deliverables, 2.1 and 2.2. The requirements will be used to steer the creation of the mockups and prototype interfaces that will form the basis of evaluation with end users.

3.1 Uncertainty in ground truth

Different users were enlisted to create the ground truth data sets used in the project. Work on improving methods for creating ground truths and on improving the ground truths themselves is ongoing. The question is what is useful to expose to the user trying to draw conclusions based on the ground truths. The main principles guiding our UI design are transparency and explanation, for example, on which ground truth data is a result based (transparency), and what do we know about the consensus of the evaluators when creating the ground truth data (explanation).

- **The ground truth data set** used to tune an analysis component should be accessible from that component.
- **A measure of evaluator consensus** should be available for each ground truth data set. The consensus measure needs to be understandable by the marine biology experts.

3.2 Uncertainty in computer vision components

The visual analysis components are based on the best ground truth information that we are able to obtain. Even if the ground truth information were perfect, this provides no guarantee for the performance of the fish detection & tracking, description, clustering and recognition components based on it. For each of the components, their outputs are captured using standard evaluation metrics and given a certainty score indicating the system confidence. In the example case of identifying an object to be a fish, the system assigns a certainty score (in the range of 0.0 - 1.0) to the potential fish object. Each potential fish object thus has an associated certainty score. While these certainty scores and evaluation metrics, as described in Deliverable 5.3, are well understood in the computer vision community, they are not necessarily understood and accepted by end users.

In the case of the fish detection component, these errors can be reduced to the under- and over-detection of fish. In other words, objects incorrectly detected as fish are false positives, and contribute to an overestimate of the number of fish; whereas fish that are not recognised as such are false negatives and contribute to an underestimate. Statistical measures for these can be provided on a per component basis. Since counting fish is the foundation for all other analyses in the system, it is essential that the user has easy access to these measures at all times when interacting with the system. The measures should be easily available in a consistent way thrughout the whole system.

- **A measure of over- and underestimating fish detections** needs to be easily available in a consistent manner at all times.

The relationship between the certainty score and the expected true and false positive rates should be understandable by end users.

- **True and false positives per certainty score** can be provided where there is sufficient ground truth data.

The certainty scores for analysis components need to be calibrated so that counts based on multiple analysis components can be combined together in some meaningful way.

- **Calibrate the certainty scores per analysis component** so that counts from multiple components can be combined.

Analysis components may produce certainty scores that do not correspond to the actual true positive percentages. For example, for a certainty score of, say, 0.8, may correspond to 93% of detected fish being indeed fish, whereas for a certainty score of 0.6 perhaps only 80% are fish. It would be easier for users if the certainty scores could be calibrated so that a score of 0.9 would indicate that 90% of the detected fish are true positives.

- **Calibrate the certainty scores** so that the values correspond to the expected true positive percentages.

A data selection contains a distribution of potential fish objects with their corresponding certainty scores. These can be used to create a certainty score profile for the specific data set[3]. A certainty score profile indicates for a specific subset of data, analysed by the same component, the differing numbers of identified fish per certainty score interval.

The certainty score profile can be visualized as the distribution of the certainty scores in 10 intervals of 0.1 for a specific data set. For example, X% of the dataset contains potential fish objects with a certainty score >0.8 and ≤ 0.9. The dataset, by definition, does not include real fish that were completely missed by the analysis component. The number of false negatives, that is the number of real fish not detected, can be calculated using the ground truth data set used to develop the analysis component. This would then give a percentage of false negatives that can be added as an "11th" column in the certainty score profile showing the number of fish that have (likely) been missed by the system. If this is extraploted to all later datasets analysed by the same component then an estimation can be made of both the true positives and the false negatives. These two together give the end user the best estimation of the true number of fish.

- **Include false negative estimates in certainty score profiles** where appropriate.

The certainty score profile may also indicate the likely false positive/false negative distribution in each of the certainty score categories.

- **Indicate levels of expected true positives in certainty score profiles**, perhaps on demand.

Datasets analysed by different components have different certainty score profiles. The certainty score profile for the specific analysis component should be easily accessible.

- **Certainty score profile for each component** needs to be easily accessible.

[3]To simplify the explanation, we assume that the data selection contains only certainty scores from a single analysis component.

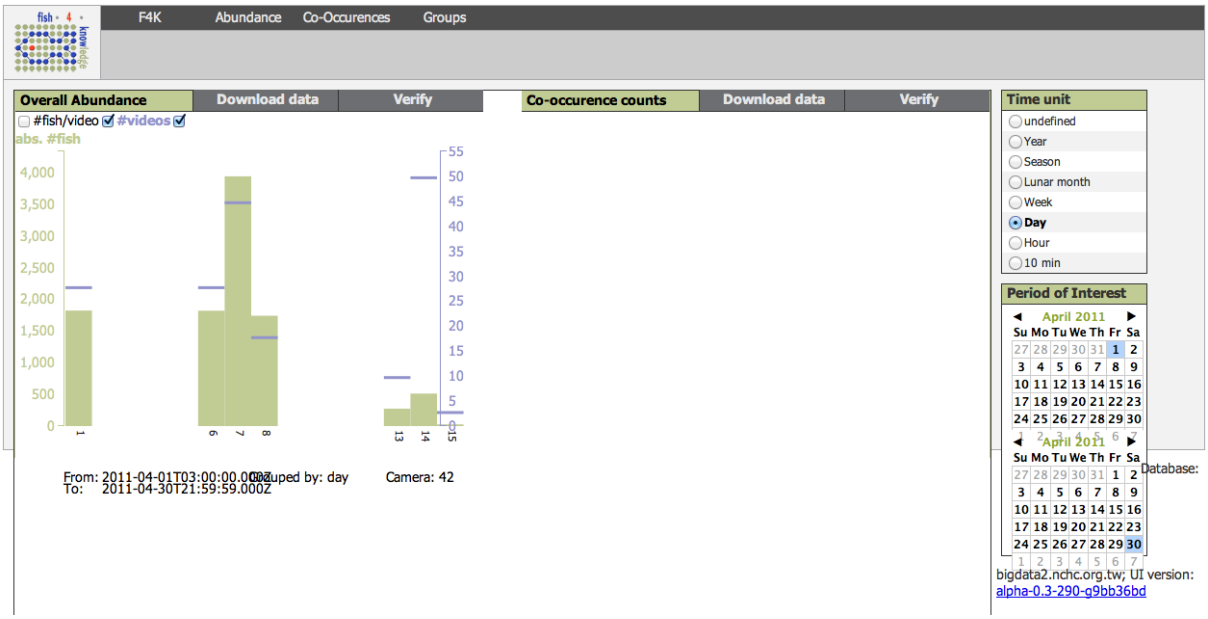


Figure 14: **Initial prototype showing raw counts of fish per video and numbers of videos analysed for the days in April 2011.**

4 Basic UIs for data visualization

Our approach to developing the interface is iterative, in that we first produced an initial "strawman" interface that could be used among the project team as a first prototype 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 14.

Having developed this initial prototype, we were able to develop our ideas on the user interface design further. This section discusses the component-based mockups that will be implemented using the data produced by the image processing components. We will discuss what data analyses can be performed with our prototypes (section 4.1), the basic functionalities of the user interface that support these analyses (section 4.2), and the mockups of our user interface designs (section 4.3).

4.1 Main data analyses

4.1.1 The population metrics and the 4 main variables

The primary analyses of the Fish4Knowledge data are based on counts of fish which can be calculated using 4 main variables: the timeframe and location of fish occurrences, and the species and behaviors of fish. The counts of fish is usually called *abundance* by marine biologists. Additionally, the *abundance in growth rate*, the *species richness*, and the *species composition* are complementary metrics for the analysis of fish counts. The *abundance in growth rate* is the rate in percentage at which the counts of fish increase in a given time period. The *species richness* concerns the number of species recognised in a population of

fish. The species richness can be calculated in counts (i.e., the counts of species recognised in a population) or in growth rate (i.e., the rate in percentage at which the counts of species increase in a given period of time). The *species composition* concern the distribution of fish for each species present in a population. The species composition can be given in counts (i.e., the set of counts of fish for each species of a given population), or in percentage (i.e., the proportion in percentage of fish from each species calculated over the total number of fish in a given population).

The 6 metrics mentioned above are the primary metrics that support the analysis of fish demographics: the *abundance* in counts of fish and in growth rates, the *species richness* in counts of species and in growth rates, and the *species composition* in counts and in percentages of fish for each species of a population. These metrics were derived from the user study we conducted and reported in Deliverable 2.1. As mentioned in section 1.3, we call these metrics the *population metrics*. We assume that marine biologists are used to perform multivariate analyses of the population metrics, which basically consist of comparing metrics' results for various timeframes, locations, species, or behaviors.

The usage of population metrics in the UI is illustrated by Fig. 15 to 20.

4.1.2 The uncertainty metrics and the 3 additional variables

On top of these primary data analyses, our system involves the analysis of the uncertainties contained in the data. Automated video analysis introduces errors in the counts of fish because some fish are not recognized (False Negatives), or because some non-fish objects are counted as fish (False Positives). As mentioned in section 1.3, we support users with *uncertainty metrics* so that they can perform the analysis of uncertainties and evaluate the levels of confidence in the patterns observed in the data. We consider 3 sets of uncertainty metrics: the *detection probabilities*, the *estimation of video analysis errors*, and the *estimation of statistical variability*.

The detection probabilities, the implied data analysis variables and the error correction mechanism

The *detection probabilities* indicate an estimation of the certainty of automatically extracted data. Users are provided with 3 detection probabilities for the detection of fish, species and behaviors. Detection probabilities can be classified in intervals of 0.1 from 0 to 1, i.e., [0.0,0.1), [0.1,0.2) ... [0.9,1.0]. For instance, a fish detected with a high detection probability would be in the [0.9,1.0] interval. This measure is independent of the detection probability of its associated species, which may lie in the range of [0.1,0.2) which denotes a high uncertainty. In this example, we are very sure that the object is a fish, but we are uncertain that the species is correct.

These metrics are called *certainty scores* in other documents related to the project. But for the User Interface, we call these metrics *detection probabilities* because we assume that this term is easier to understand for marine biologists.

These detection probabilities can be used as a threshold for selecting the fish to take into account in the calculation of counts of fish, growth rate or any population metric. For instance, marine biologists could wish to deal only with highly certain recognition of species, and set the *species recognition probability* threshold to 0.9. Fig. 27 gives an example of the usage of detection probability thresholds.

The detection probabilities add 3 more variables to the multivariate data analyses that can be performed on population metrics. Thus multivariate data analyses can be performed with up to 7

variables that are attached to each detected fish: timeframe, location, fish detection probability, species, species recognition probability, behavior, and behavior recognition probability.

Additionally to detection probability thresholds, we support a simple mechanism for visualizing the probability of errors implied in the detection of fish, species and behaviors. It consists of visualizing population metric's results that integrate the correction of the estimated errors. This error correction mechanism is applicable for population metrics that are based on counts of fish (i.e., abundance and species composition), and is not applicable to the study of species richness. It basically consists of multiplying the counts of fish by the detection probabilities implied in the multivariate analysis, in order to obtain corrected counts of fish. Fig. 32 gives an example of the visualization of errors denoted by detection probabilities. The calculation of a corrected count of fish consists of the following steps:

1. Select the fish population that respects the multivariate criteria (e.g., timeframe, location, species...).

2. If a threshold criterion is applied on fish detection probability, select subsets of fish population that belong to each fish detection probability bin (e.g., a subset of fish with a fish detection probability of 0.1, one for fish with a probability of 0.2, of 0.3...).

3. If a threshold criterion is applied on species recognition probability, for each subset selected in the previous step, divide the subset in further subsets of fish that belong to each species recognition probability bin (e.g., fish with a species recognition probability of 0.1, 0.2, 0.3...).

4. If a threshold criterion is applied on behavior recognition probability, for each subset selected in the previous step, divide the subset in further subset of fish that belong to each behavior recognition probability bin (e.g., fish with a behavior recognition probability of 0.1, 0.2, 0.3...).

5. For each subset selected in the previous steps, count the number of fish and multiply this count by all implied detection probabilities (i.e., the fish detection probability, the species recognition probability or the behavior recognition probability if applied as a selection criterion).

6. Add all the counts of each subsets calculated in the previous step. We obtain a corrected count of fish.

The estimation of video analysis errors and the related error correction mechanism

Specific video analysis components detect fish, recognize the species of fish, and recognize the behaviors of fish. For each of these video analysis tasks, the performance of our system is evaluated with respect to a ground-truth dataset. This evaluation provides the counts of elements (i.e., fish, species or behaviors) that were correctly identified (True Positives), the counts of elements that were not identified (False Negatives), and the counts of elements that were identified but that do not correspond to any real elements (False Positives). We can use this evaluation to estimate the number of True Positives, False Negatives and False Positives that are likely to be contained in any fish population.

We support a simple mechanism for visualizing the probability of errors implied by the automated video analysis. It consist of calculating the rates of True Positives, False Negatives

and False Positives that are likely to be contained in every fish population, and correcting the counts used in the population metrics accordingly. For instance, given the following video analysis evaluation: the automatic detection of fish in the ground-truth dataset contained 80% of True Positives and 20% of False Negatives, and should have contained 30% more of missing False Negative fish. In other terms, 110 fish are contained in the ground-truth, 80 are correctly detected (True Positives), and 20 of detected fish are non-fish objects (False Positives), and 30 fish were not detected (False Negative). Thus the detected fish contained 20% of False Positives, and missed an additional 30% of False Negative fish. And the overall corrected number of fish should contain +10% fish. In that case, given another population of 200 detected fish, we can inform users that i) 20% of the fish are False Positives, i.e., discard 40 fish, ii) 30% of the fish are missing False Negatives, i.e., add 60 fish, and iii) the corrected count of fish is 220 fish. We aim at supplying users not only with corrected counts of fish, but also with the detailed estimation of True Positives, False Negatives and False Positives.

The recognition of each species and each behavior imply their own dedicated error rates, and thus the estimation of video analysis errors is more complex than the correction of detection probabilities errors. We assume that this estimation of video analysis errors is not relevant for the study of species richness. Note that the abundance in growth rate is not affected by this error correction mechanism. Fig. 31 gives an example of the correction of the video analysis errors.

Estimation of statistical variability

Our envisaged tool supplies 3 types of statistical measurements of variability and the related common ways to visualize them: i) the standard deviation visualized with error bars, ii) the inter-quartile range visualized with box plots, and iii) the decomposition in sub-samples of data visualized with scatter plots. All these statistics are using sub-samples of the data used for the calculation of a population metric. Users can choose how the data should be sampled. The data can be sampled per time unit, per location, per species, or per detection probability bin. For instance, if users are visualizing counts of fish per week, they can choose to calculate standard deviation with all counts of fish per days of each week. They can also choose to calculate the standard deviation for counts of fish calculated for each hour of the week. They might also observe a greater variability for counts sampled per hour rather than per days. Fig. 28 to 30 give examples of the usage of statistical variability metrics.

4.1.3 The species abundance thresholds as an extra variable

As mentioned by marine biologists during our user study, and as reported in the Deliverable 2.1, the calculation of population metrics can be done using a species abundance threshold. A species abundance threshold consists of selecting fish species for which a certain number of individual were detected. The species abundance threshold can be defined in counts (e.g., the species is discarded if less then 5 fish were detected), or in percentage w.r.t. to the overall count of fish regardless of the species (e.g., a species is discarded if it represents less than 2% of all the fish in the population).

The species abundance threshold is primarily used to discard species that do not contain a sufficient number of individual for the species to be suitable for a statistically valid data analysis. For instance, a species containing less than 5 fish should be discarded of the calculation of species richness, because there is a too high chance on the 5 fish not to belong to the species, or on the species not to constantly live in the area of study. This contributes to the support provided for users to deal with the uncertainty contained in the data. Additionally, the species

abundance threshold can also be used to study species from specific abundance range, e.g., to study only abundant, common, occasional or rare species. This is done by defining a range of abundance of interest (e.g., to study species that contain more than X fish or X percents of fish, or an interval of abundance).

The detection probabilities add one more variables to the multivariate data analyses that can be performed on population metrics. Thus multivariate data analyses can be performed with up to 8 variables that are attached to each detected fish: timeframe, location, fish detection probability, species, species recognition probability, behavior, behavior recognition probability, and species abundance threshold. Fig. 19 and 20 give examples of the usage of species abundance thresholds.

4.1.4 Data analysis tasks

We assume that biologists will perform data analysis by basically making only one variable vary at a time, so that they have a consistent scope of comparable population metrics' results. For instance, a biologist studying species X would calculate counts of fish from species X for each location, but for the same period of time and regardless of behaviors. If she also wants to study the evolution of the population over time, she might repeat the calculation of sets of counts for each period of interest.

Biologists might need to perform more variations of the populations metrics. For instance, they might study the break down of counts for various behaviors, and they might perform more complex study of growth rates. Regarding the resources available for the project, the user interface can not integrate the whole range of possible data analyses as suggested by the examples above. Thus we aim at supporting basic data analyses of population metrics. More advanced data analyses should be performed using other tools, such as those already in use in their regular working environment (e.g., R, matlab, etc.).

Our tool will support i) the identification of interesting variations in populations metrics, and ii) the evaluation of the level of confidence in the population metrics w.r.t. the potential errors of automated video analysis. The main data analyses tasks that can be performed using our user interface are:

- **Task 1: Requesting an overview of a consistent set of population metrics** where only one variable varies (e.g., counts for each month of the year and for the same location, or counts for each location but for the same period of time, etc...). We call that variable the ***x-axis variable*** because it defines the x-axis to use in the visualized graph.

- **Task 2: Comparing 2 sets of population metrics** that can be obtained through Task 1 above. The 2 sets of population metrics are of the the same metric (e.g., they are both abundance in growth rate), and they both use the same type of unit for their *x-axis variable* (e.g., both are weekly counts of fish, or both are daily counts). The 2 sets of population metrics also share the same variable parameters (amongst timeframe, location, fish detection probability, species, species recognition probability, behavior, and behavior recognition probability), except one variable which value is different between the 2 sets of metrics (e.g., they both count fish for each month of the year, but one set is evaluated for 2011 and the other for 2012). Figures 21 and 25 give examples of the visualization of 2 comparable sets of population metrics.

- **Task 3: Requesting a view of the potential errors** involved in the provided population metrics. This task involves the uncertainty metrics mentioned in section 4.1: the *detection probabilities*, the *estimation of video analysis errors*, and the *estimation of statistical variability*. Additionally, this implies additional explanations provided for users to understand the underlying computational layers that produced the visualized data, and the related errors and uncertainty introduced in the data. Figures 27 and 32 give examples of the visualization of uncertainty metrics.

4.2 Basic user interface functionalities

This section discusses the user interface functionalities needed for the 3 data analysis tasks mentioned above.

4.2.1 Task 1: Request a consistent set of population metrics

To allow users to perform the Task 1, i.e., requesting a set of population metrics where only one variable varies, the user interactions supply 3 main functionalities:

- F1: Support the selection of the population metric of interest.
  - Display all available population metrics, amongst *abundance* in counts or growth rates, *species composition* in counts or percentages, and *species richness* in counts or growth rates.
  - Allow the selection of a population metrics to calculate.
  - Indicate the selected population metrics.

  Figures 15 to 20 illustrate the selection of the population metric of interest.

- F2: Support the calculation of the set of population metrics of interest.
  - Display all available variables and variables' values.
  - Allow the selection of variables' values or sets of values to select the fish population of interest.
  - Indicate the selected values or sets of values.
  - Allow the selection of the *x-axis variable* that defines the set of population metrics to calculate. The population metric will be calculated for each of the values selected for that variable.
  - Indicate the selected *x-axis variable*.
  - Display the set of results of the population metrics calculated for the selected set of *x-axis variable*'s values, and for the selected fish population of interest.

  Figures 22 to 24 illustrate the selection of the variables' values of interest.

4.2.2 Task 2: Compare two sets of population metrics

To allow users to perform the Task 2, i.e., comparing 2 sets of population metrics's that can be obtained through Task 1, the user interactions supply the following functionalities.

- F1: Support the calculation of two comparable sets of population metric, on the basis of a set of population metric that was previously calculated
  - Allow the selection of one alternative variables' values for which another set of population metrics must be calculated
  - Indicate the selected alternative values
  - Overlay the 2nd set of results of the population metrics on top of the initial set of results

  Figures 21 and 25 give examples of the visualization of 2 comparable sets of population metrics.

4.2.3 Task 3: Request an overview of the uncertainties

To allow users to perform the Task 3, i.e., requesting an overview of the potential errors involved in the provided population metrics, the user interactions supply the following functionalities.

- F1: Provide explanations of the data processing steps and the nature of the errors that each step can introduced.
- F2: Support the usage of the uncertainty metrics of interest
  - Display all available uncertainty metrics
  - Allow the selection of the uncertainty metrics to study
  - Display the results of the selected uncertainty metrics

  Figures 27 to 32 give examples of the usage of uncertainty metrics.

4.3 Preliminary user interface mockups

Figures 15 to 32 show the user interface mockups we have designed, which support our reflections and experimentations for the user interface of the system in the project. Figures 15 to 25 give examples of the usage of population metrics. Figures 26 to 32 give examples of the usage of uncertainty metrics.

Using Figure 15 as an example, we briefly explain the design of the UI. Figure 15 shows an example of multivariate analysis where only one variable is usable with interactive widgets: the timeframe of interest. The other variable widgets are available through the *"Parameters"* menu on the left side of the *Zone D*. The *fish detection probability* variable is also used to select the fish population of interest, and is set to a range of [0.7, 1] as indicated in the title of the graph. The other variables are set on default values, i.e., all possible values are selected. The *x-axis variable* is set to the month of year, which is indicated by the icon overlaid on the related variable widget in *Zone D*.

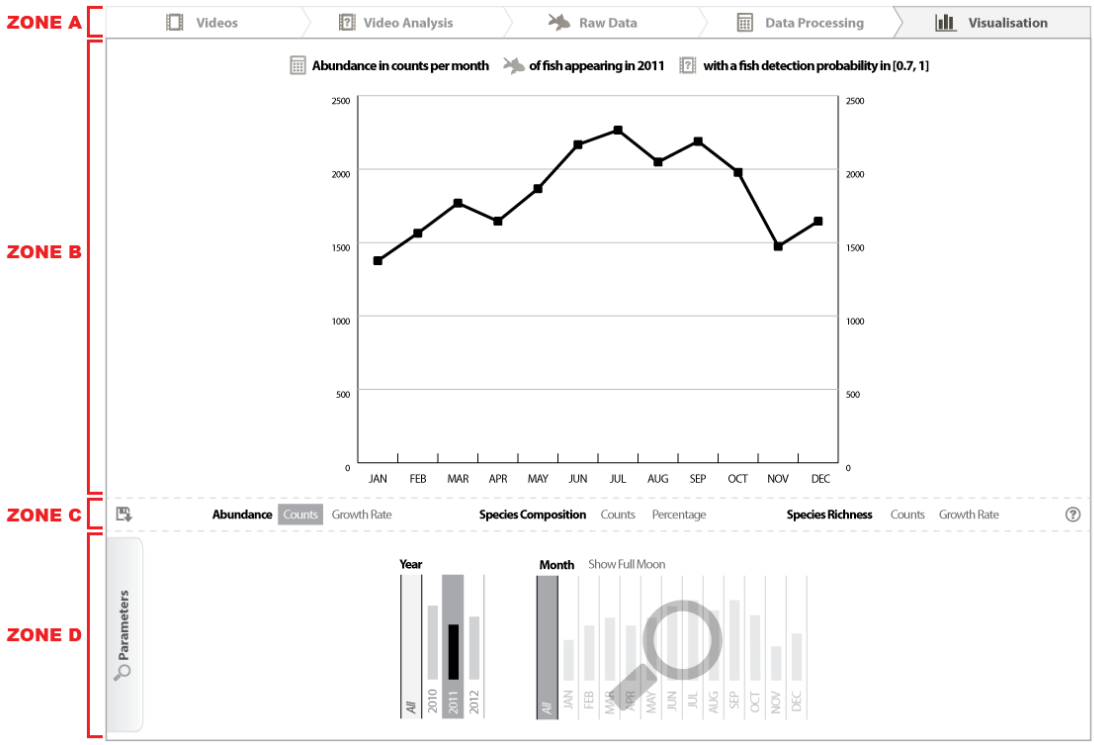


Figure 15: **Counts of fish over the month of the year** (i.e., the *abundance in counts*).

The UI is organized in 4 zones. The *Zone A* contains a menu to access explanations about all the layers of computations that produced the current visualization. This support the functionality F1 of Task 3 described in section 4.2.3. The *Zone B* contains the visualization of the selected population and uncertainty metrics. The title of the visualization is automatically generated. It describes the metrics that are visualized, and all the related variables implied in the calculation of the metrics. The *Zone C* contains a menu of all available population metrics. This supports the functionality F1 of Task 1 described in section 4.2.1. The zone also contains a help button (on the right side) and a download button (on the left side) to get the raw numerical data that are displayed on the graph, in the form of a csv file. The *Zone D* contains the interactive widgets for the selection of the variables used to calculate the population metrics, and for the selection of the uncertainty metrics to calculate. This supports 3 functionalities described in section 4.2: F2 of Task 1, F1 of Task 2, and F2 of Task 3. The interactive variable widgets are not displayed at all times, because it would clutter the display space and prevent users to focus on useful variables. Thus we provide a menu to select the widgets to display. The usage of the widget menu is described in Fig. 23. The variable widgets contain a small graph providing an overview of the visualization that would be generated if the variable is selected as the *x-axis variable*.

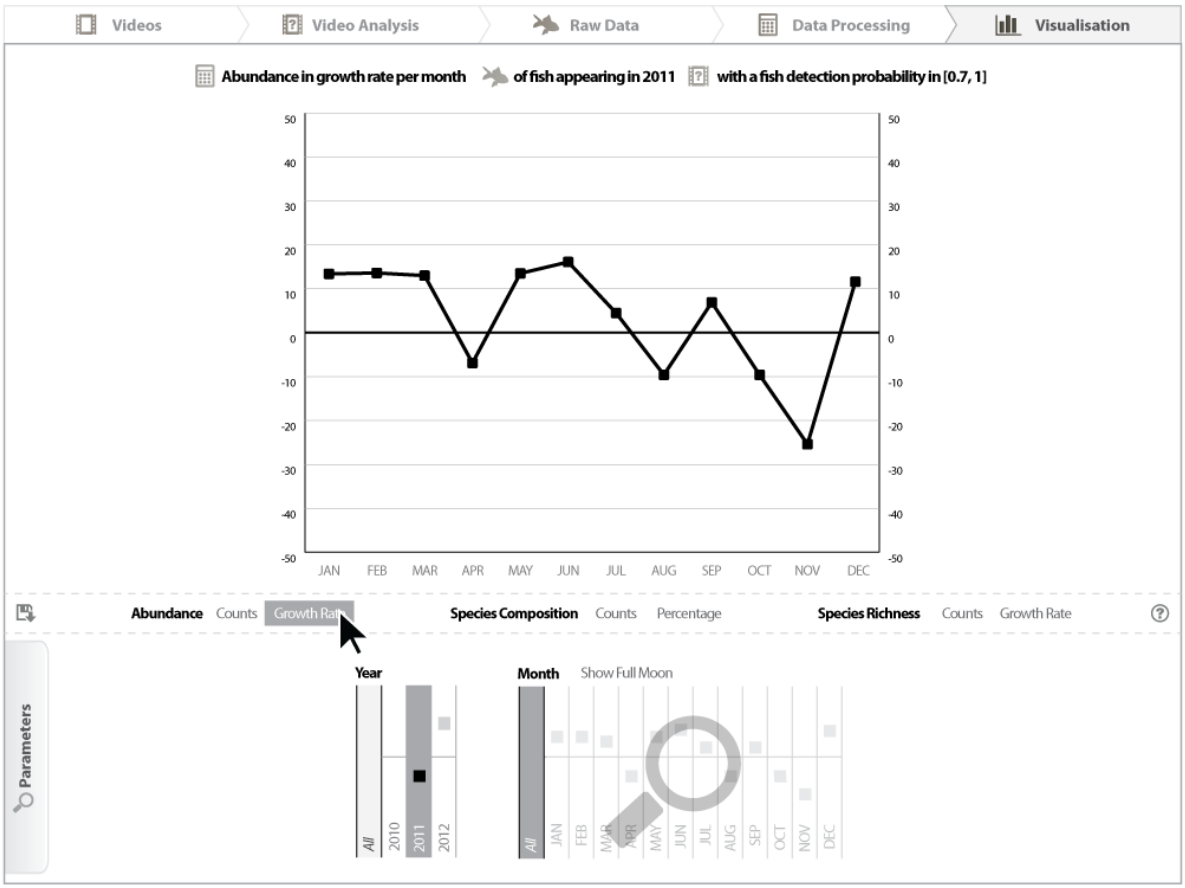


Figure 16: ***Abundance in growth rate*** for each month of 2011.

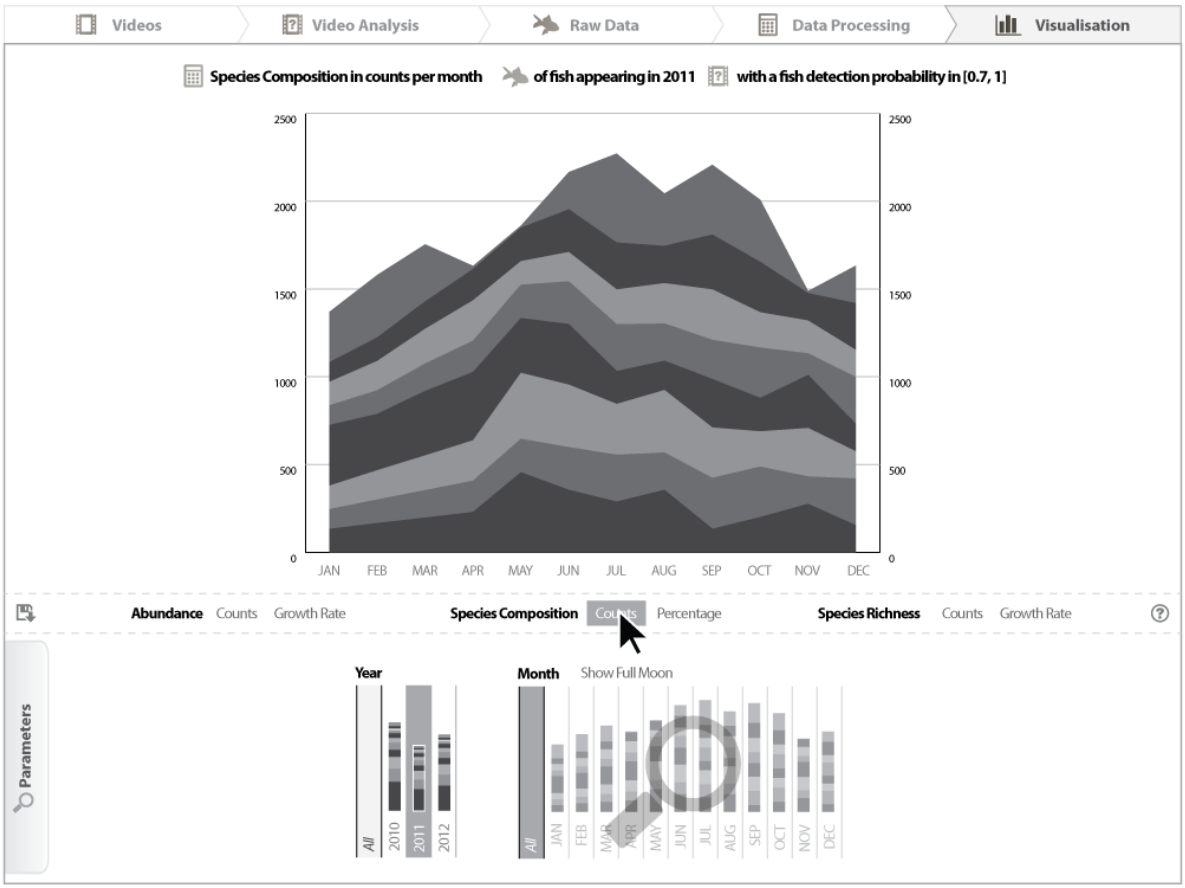


Figure 17: ***Species composition in counts*** for each month of 2011.

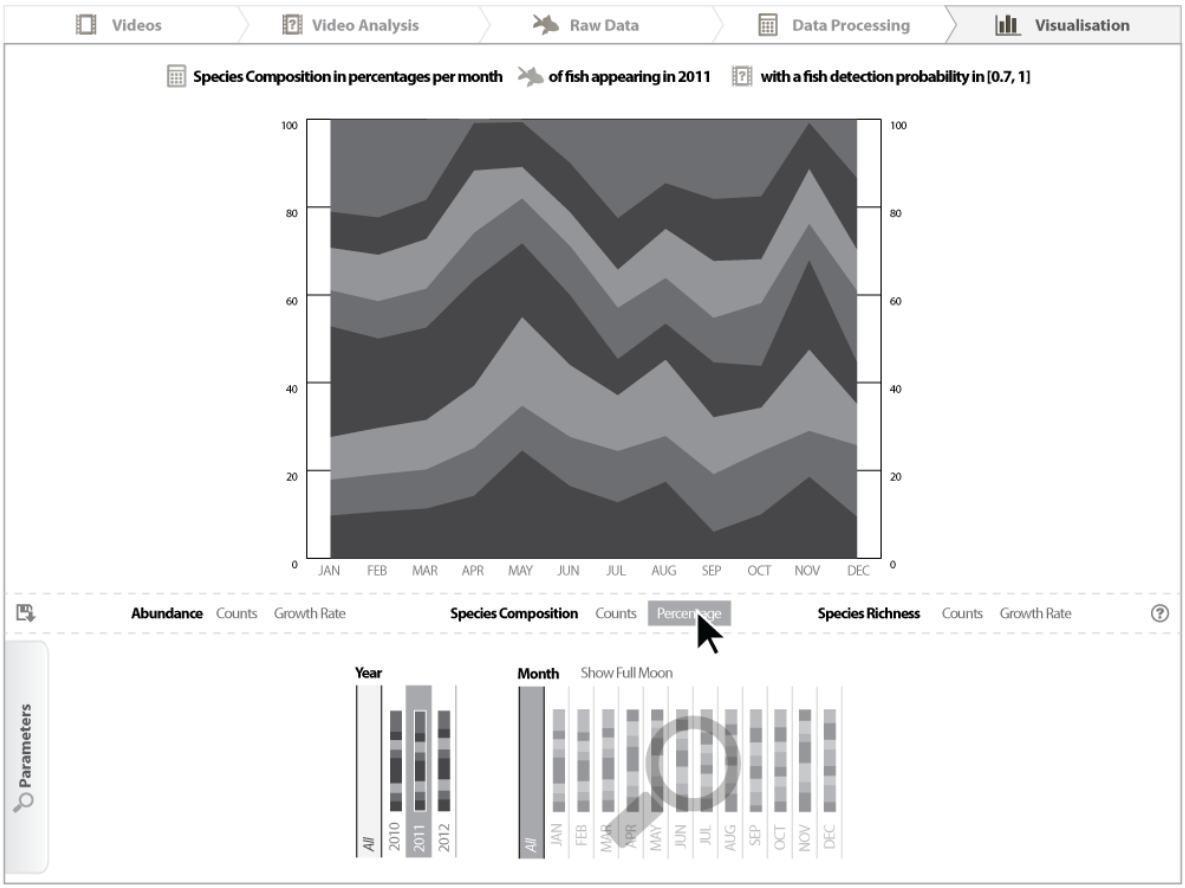


Figure 18: ***Species composition in percentages*** for each month of 2011.

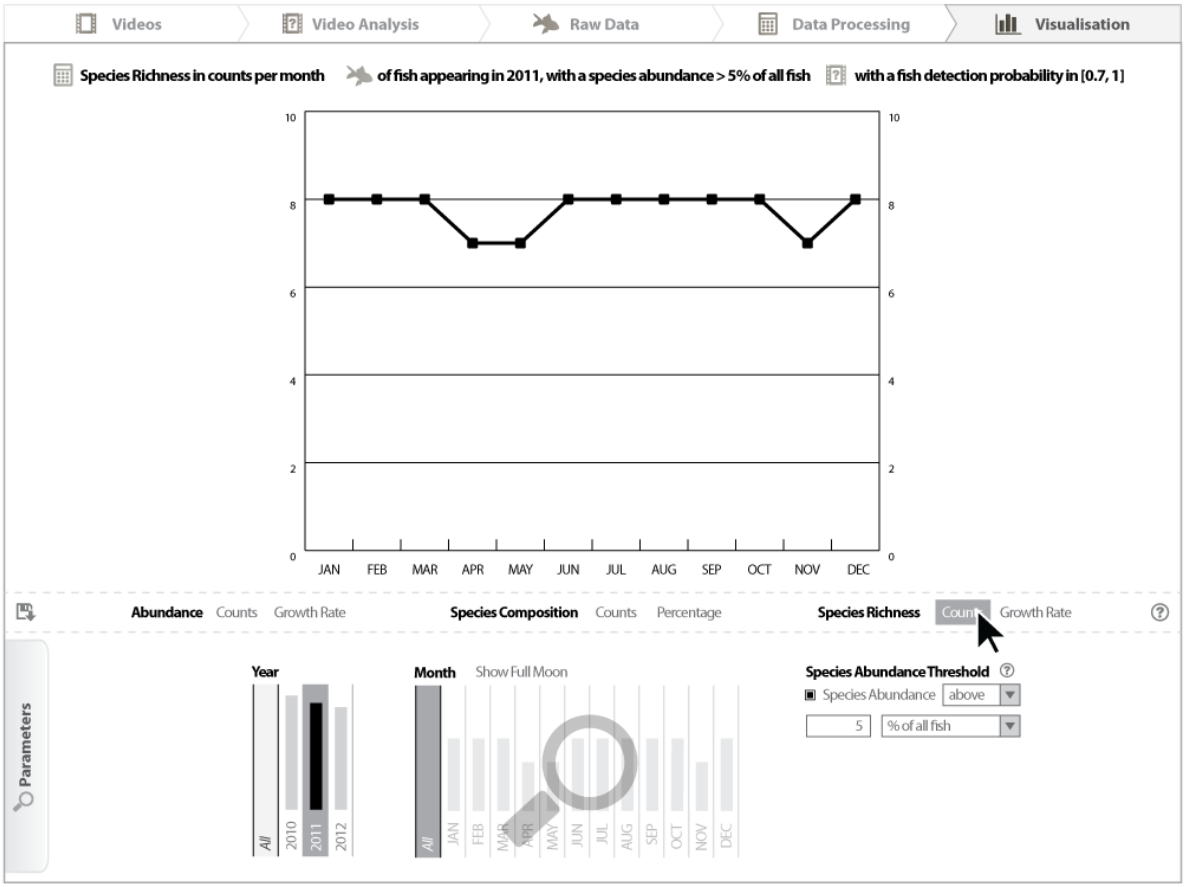


Figure 19: ***Species richness in counts*** for each month of 2011. It uses a *species abundance threshold*, as defined in section 4.1.3.

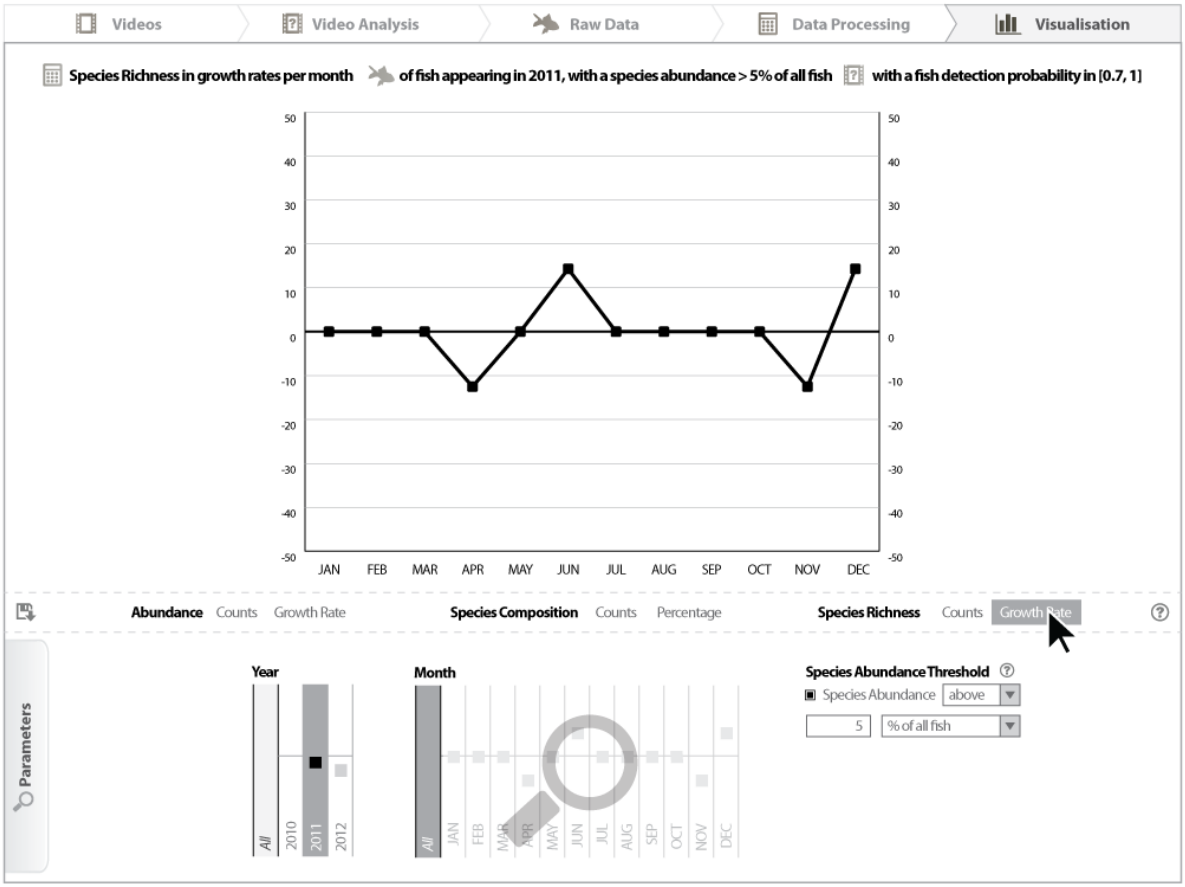


Figure 20: ***Species richness in growth rates*** for each month of 2011. It uses a *species abundance threshold*, as defined in section 4.1.3.

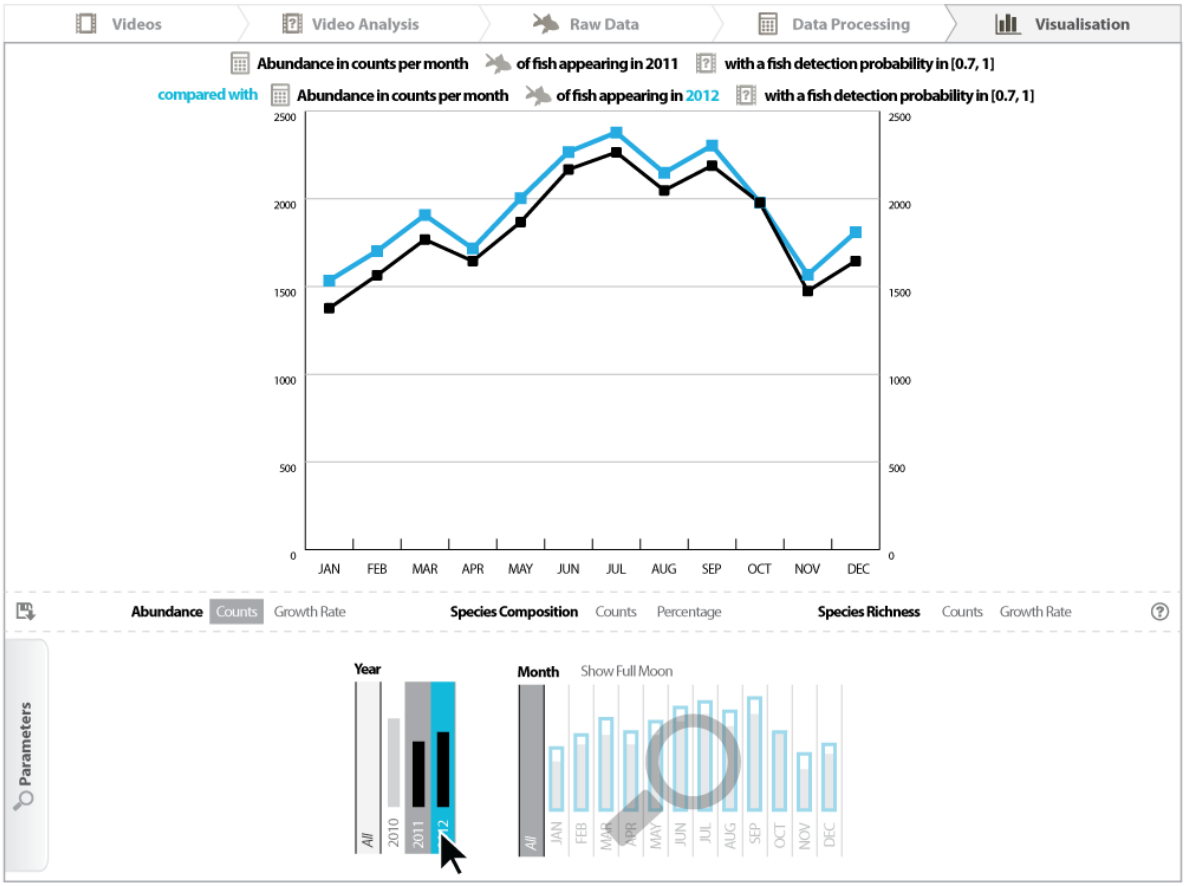


Figure 21: **Two comparable sets of population metrics**, i.e. the *abundance in counts* for each month of 2011 and 2012. The alternative set of population metrics is obtained by a rollover on the alternative variable value. In this example, the user rolls over the year 2012 in the *"Year"* widget on the left. The alternative variable value is highlighted in blue. A new set of population metrics is calculated using all the other variables used for the previous set of metrics (e.g., the fish detection probability is within a range of [0.7, 1.0], and the *x-axis variable* is the month of the year). The new set of population metrics is displayed in blue in the main graph. The title of the graph describes the 2 sets of population metrics that are compared. This supports the functionality F1 of Task 2 described in section 4.2.2.

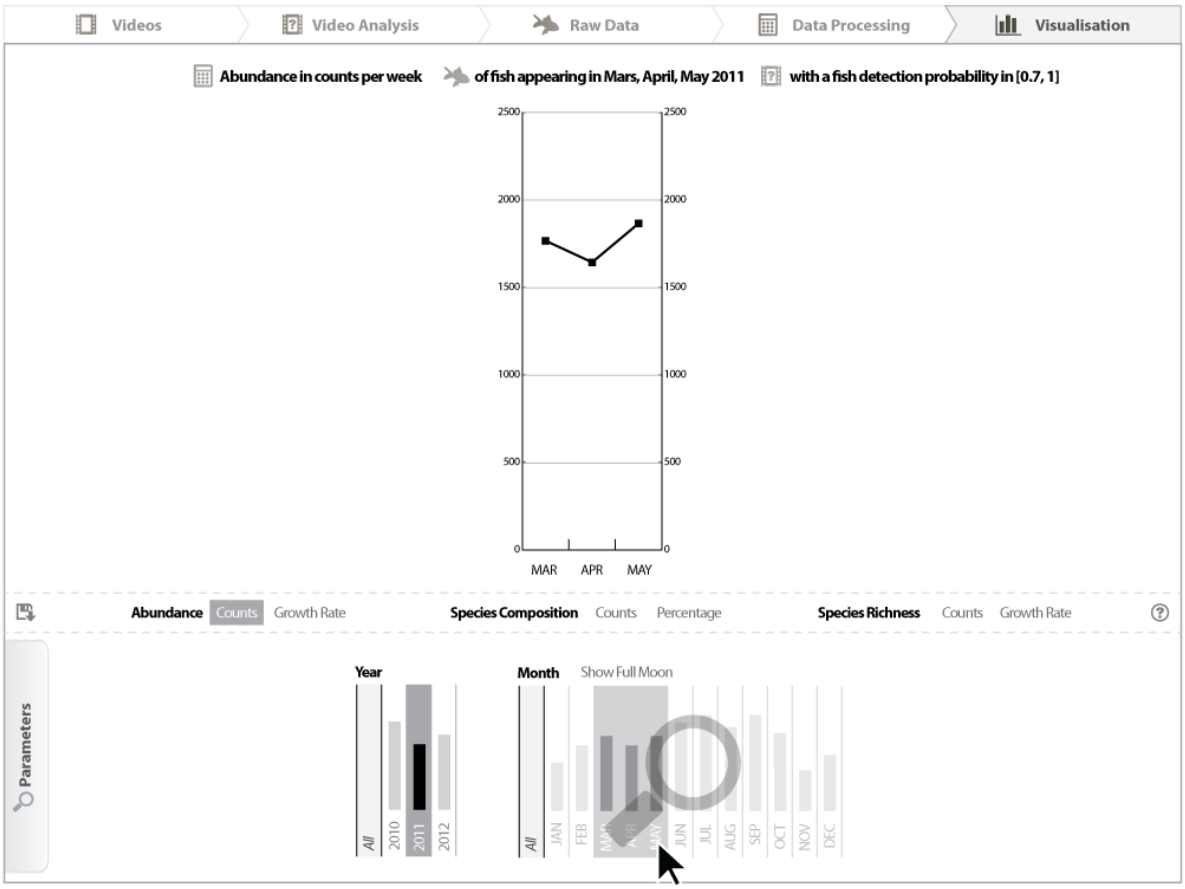


Figure 22: **Selection of variable values.** The user started with the variables of the data analysis shown in Fig. 15 and 21 above. The initial timeframe of interest was the whole year 2011. It this figure, the user has narrowed down the timeframe to the months of March, April and May 2011. This is done by clicking on each of the months in the *"Month"* widget (in the bottom of the UI).

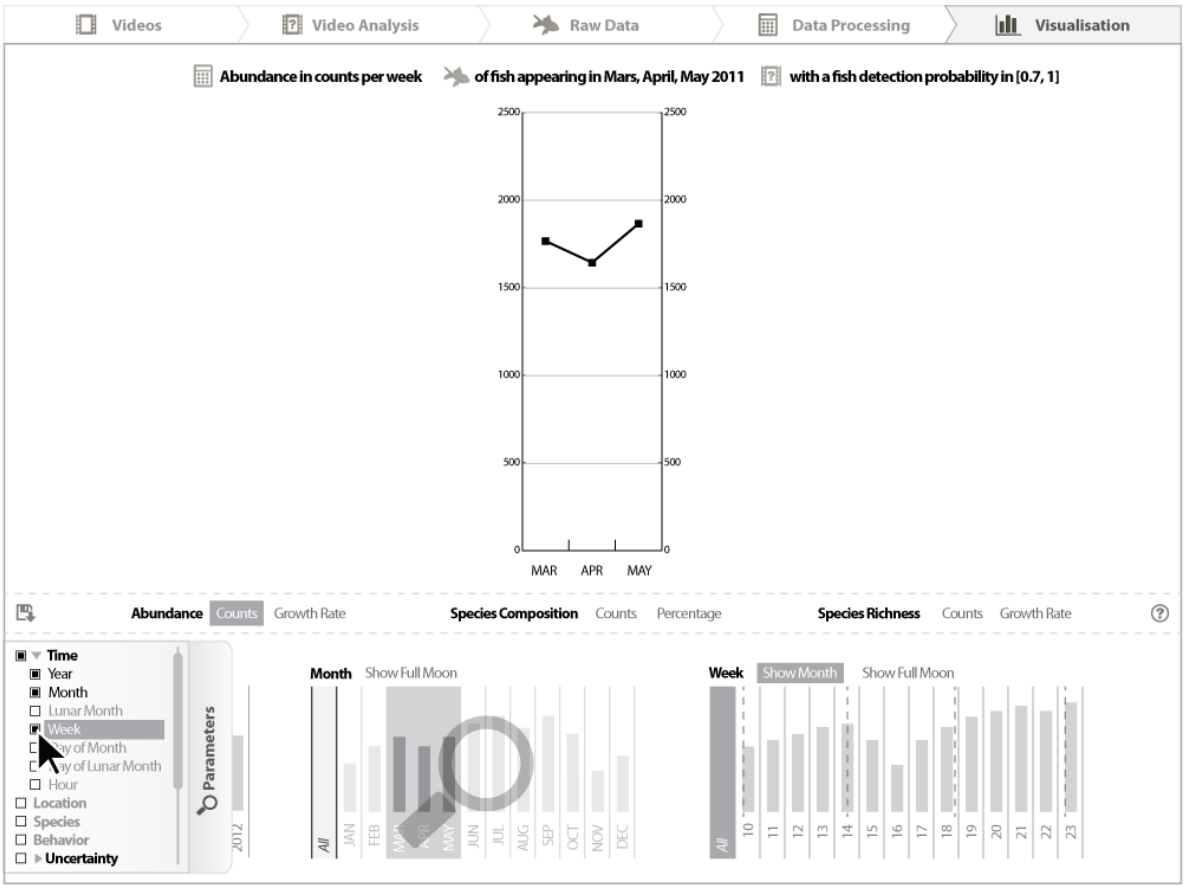


Figure 23: **Selection of parameters.** The parameter menu gives users access to the interactive widgets used i) to select the variables for the calculation of population metrics, and ii) to select the uncertainty metrics to display. In this example, the user requests the display of the *"Week"* widget that allows the selection of the weeks of interest.

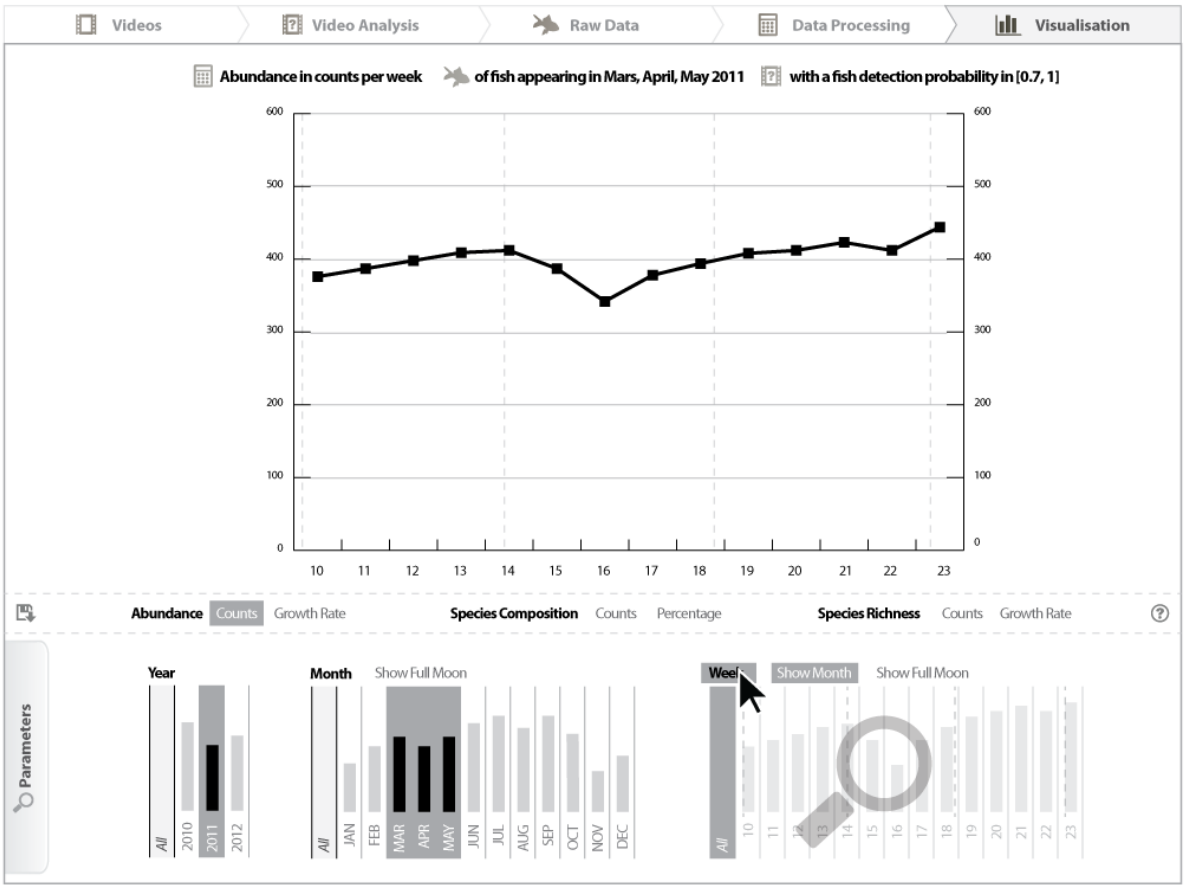


Figure 24: **Selection of the *x-axis variable*.** In this example, the user sets the x-axis to weekly counts of fish. This is done by clicking on the title of the *"Week"* widget (on the bottom right of the UI). The x-axis displays the week numbers for each week of the timeframe of interest (e.g., March, April and May 2011).

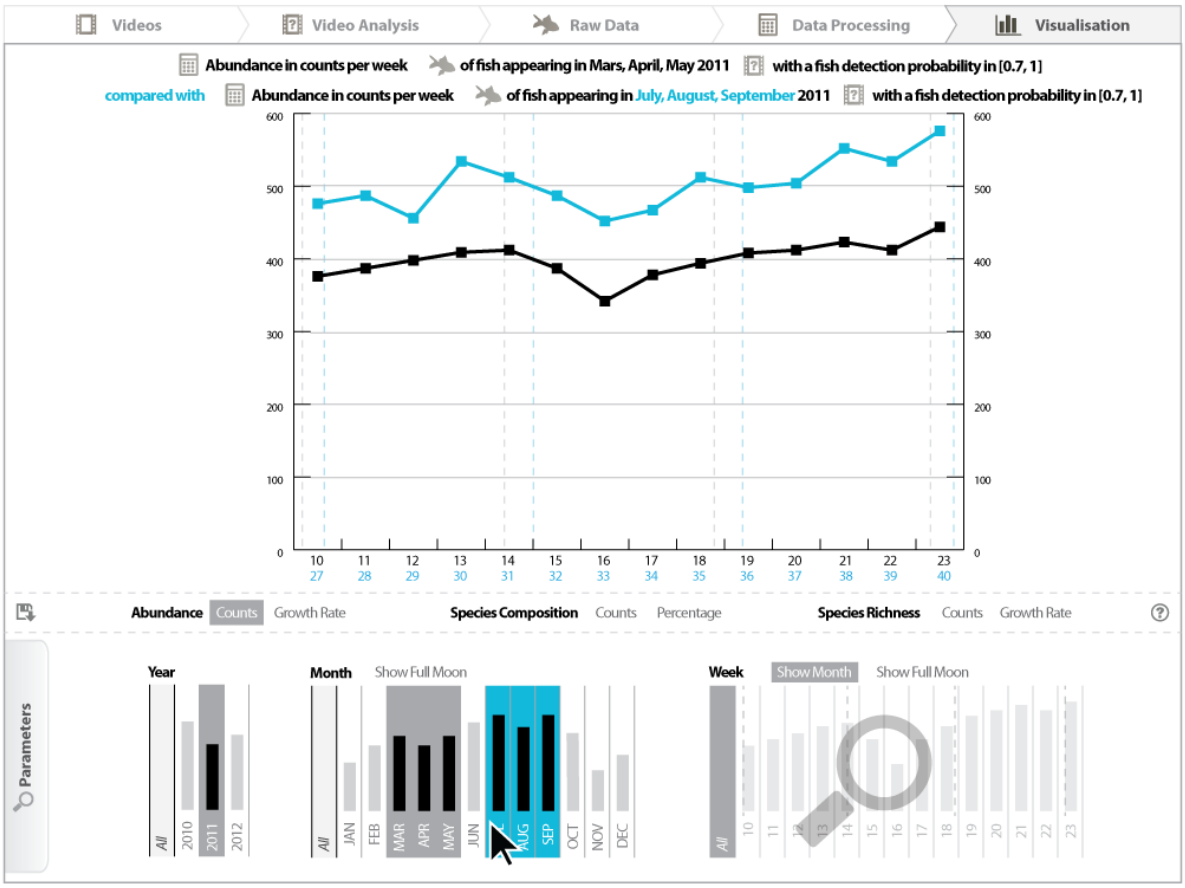


Figure 25: **Comparison of 2 sets of population metrics over consistent periods of time.** In this example, the user rolls over the month of July in the *"Month"* widget. This triggers the calculation of a new set of population metrics, overlaid in blue in the main graph. This new set of population metrics is calculated over the same duration but with a different start date, and uses the same time window as the previous set of population metrics (e.g., 3 months). In this case, the user rolls over only 1 single month (e.g., July), but the set of population metrics is calculated over 3 consecutive months.

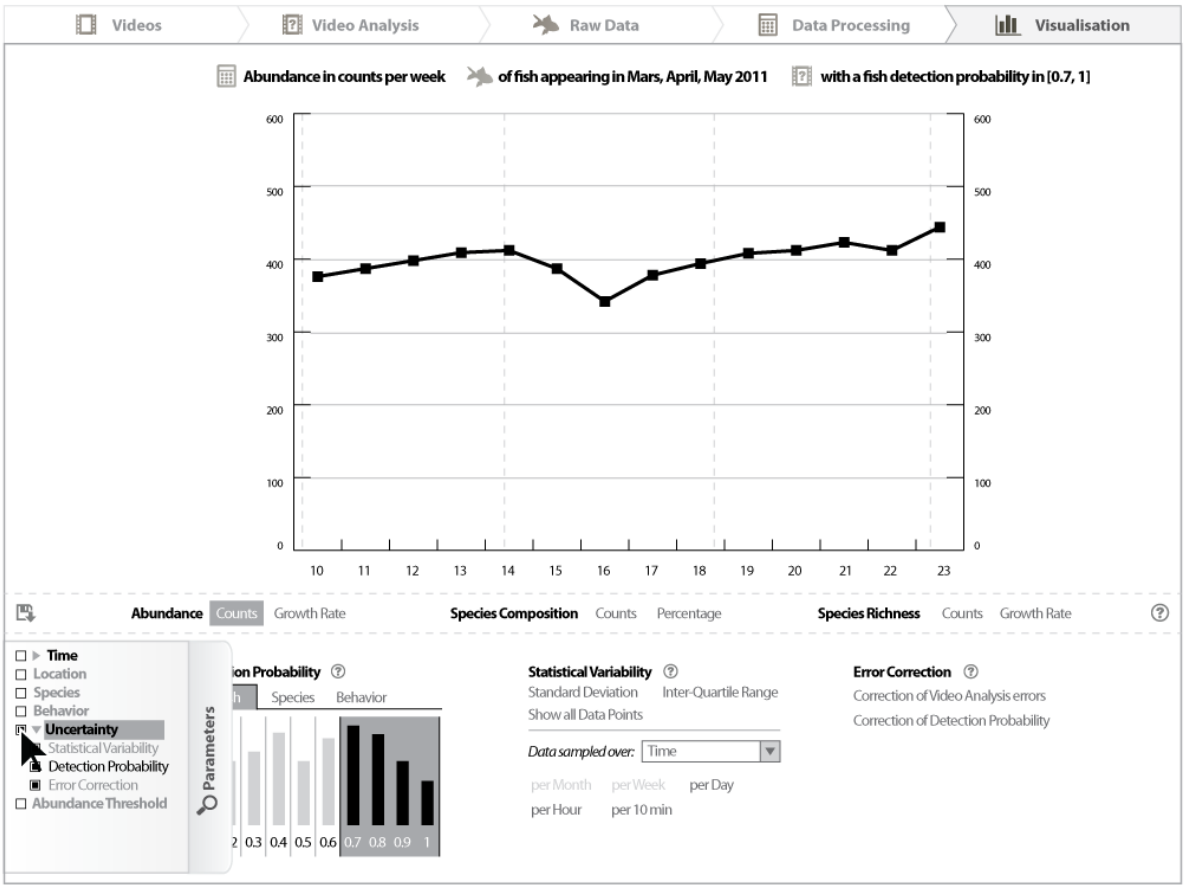


Figure 26: **Uncertainty metrics provided in the parameter menu.**

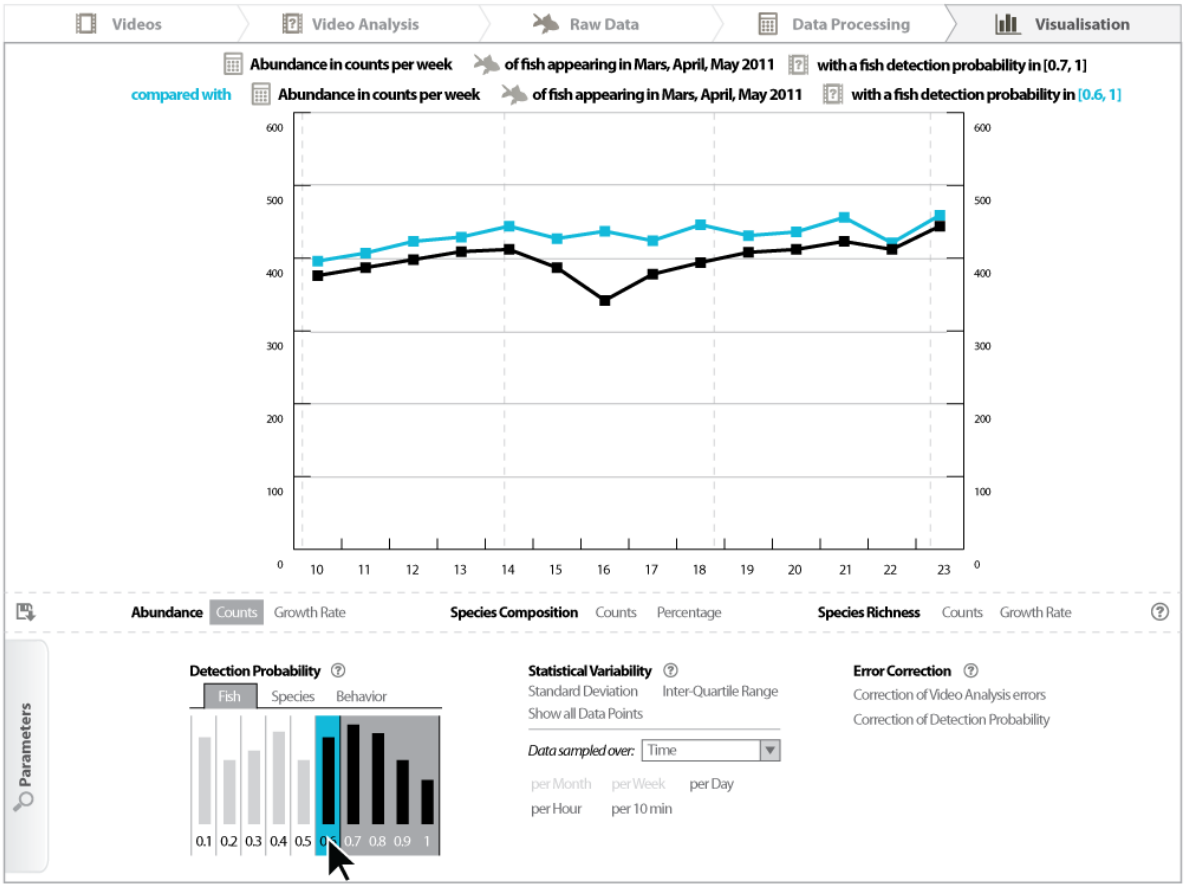


Figure 27: **Usage of *detection probability thresholds*.** In this example, the fish detection probability variable is set to the range [0.7, 1]. The user rolls over the 0.6 fish detection probability. It triggers the calculation of a new set of population metrics for a fish detection probability within [0.6, 1]. The new set of population metrics is displayed in blue in the main graph.

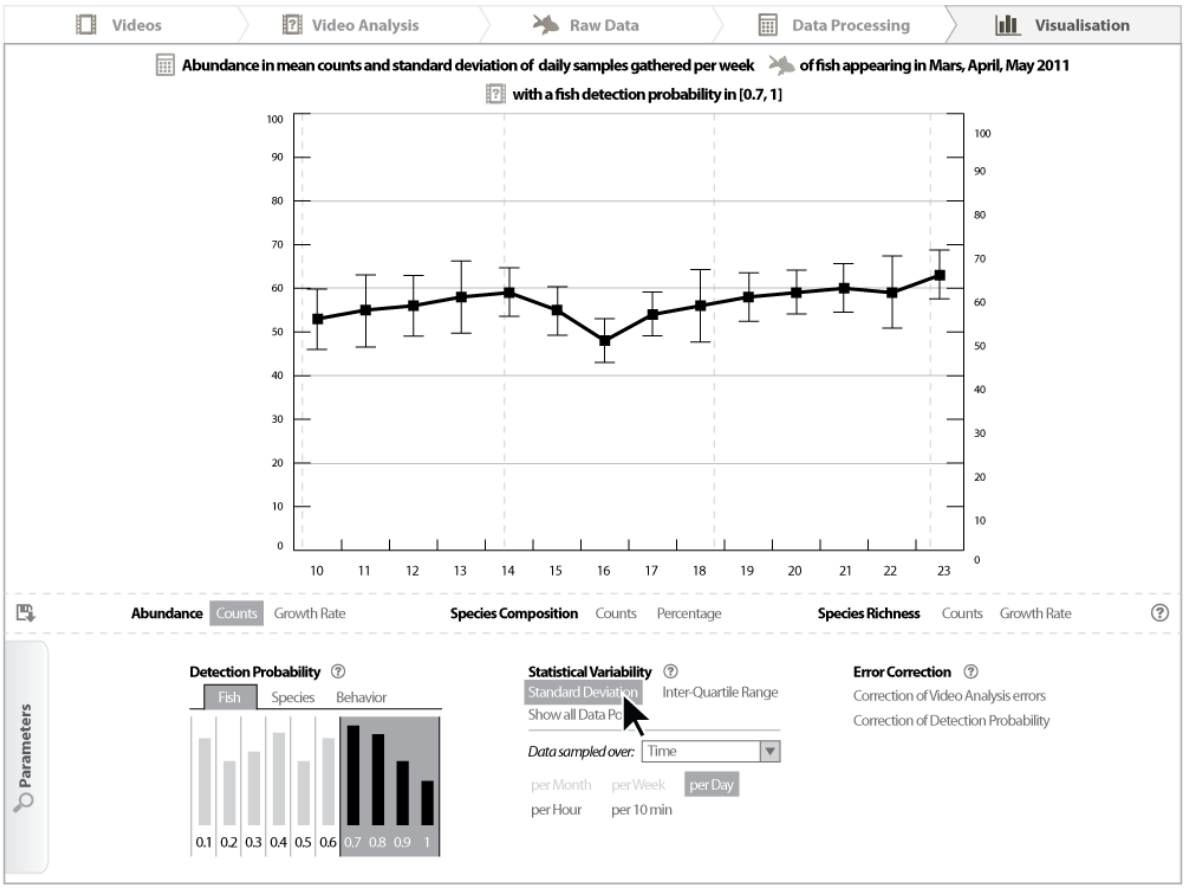


Figure 28: **Usage of *statistical variability over time*, and the display of *standard deviation*.** In this example, the users chooses to calculate the standard deviations for data sampled per days of the week, for each week of interest. The population metric is calculated for each day of the week, for each week of interest (e.g., weeks 10 to 23). For each week, the standard deviation is calculated using the daily data samples.

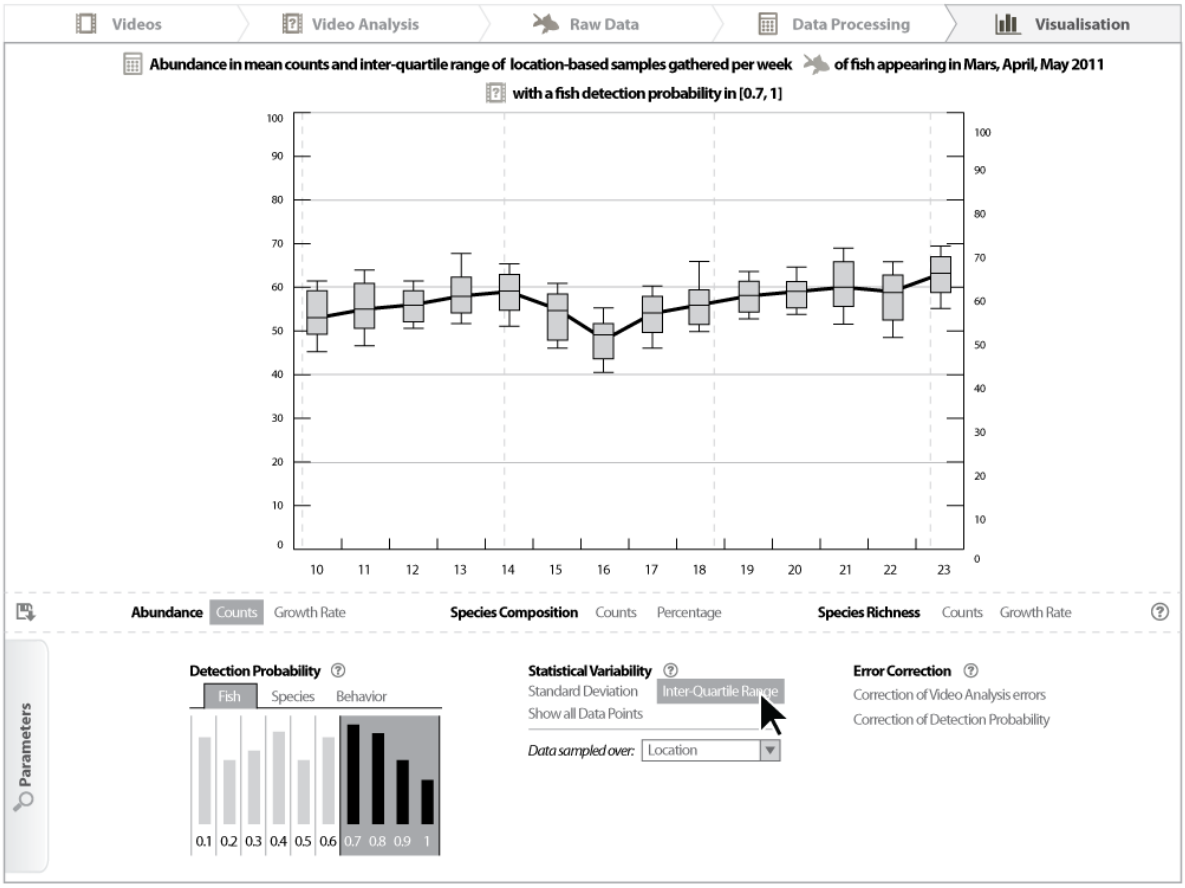


Figure 29: **Usage of *statistical variability over location*, and the display of *inter-quartile range*.** In this example, the user chooses to calculate the inter-quartile ranges for data sampled per location. For each week of interest (e.g., weeks 10 to 23 in year 2011), the population metric is calculated for each location, and the inter-quartile range is calculated using the data sampled for each location. This shows the variability over location, for each week of interest.

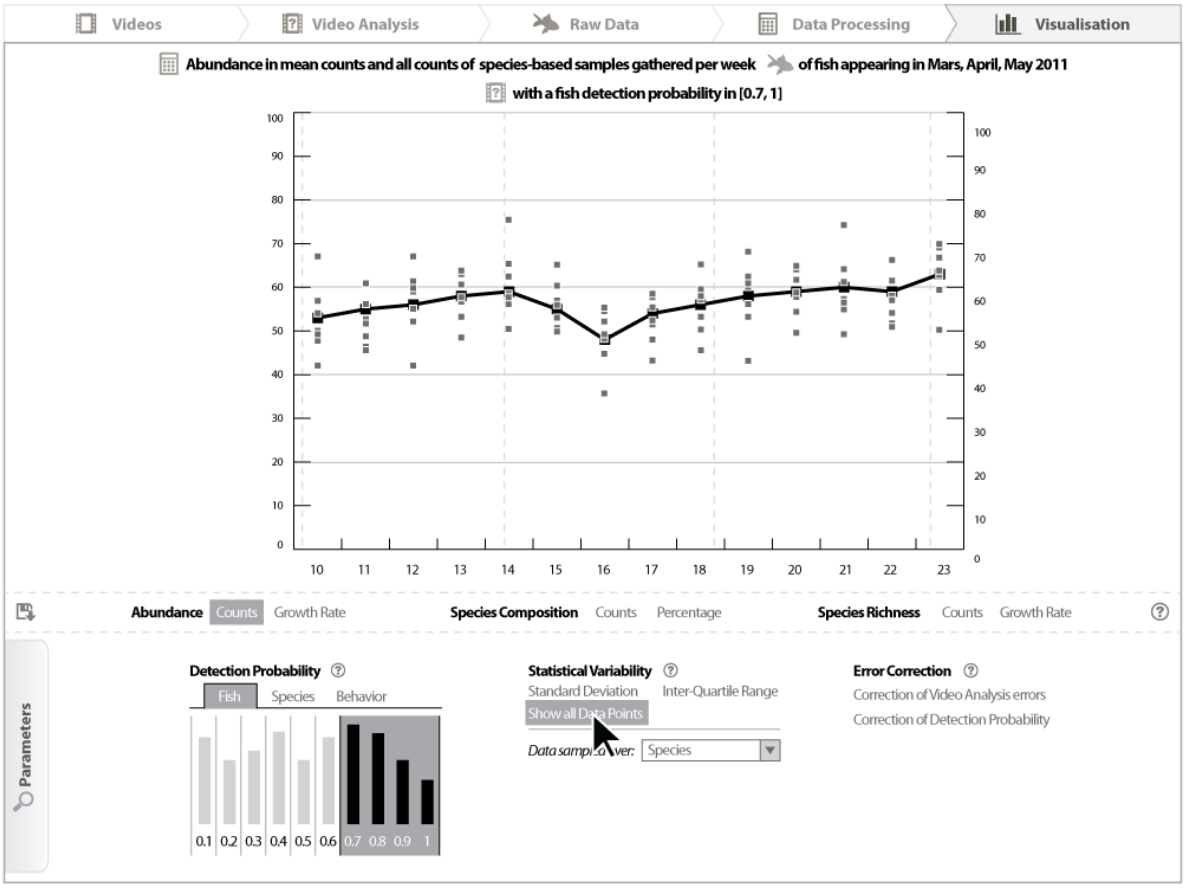


Figure 30: **Usage of *statistical variability over species*, and the display of *all data points*.** The user chooses to display all the data points for data sampled for each species of the population. For each week of interest (e.g., weeks 10 to 23 in year 2011), the population metric is calculated for each specific species (e.g., weekly counts of fish from species X, species Y...). The graph displays the values obtained for each species. This shows the variability over fish species, for each week of interest.

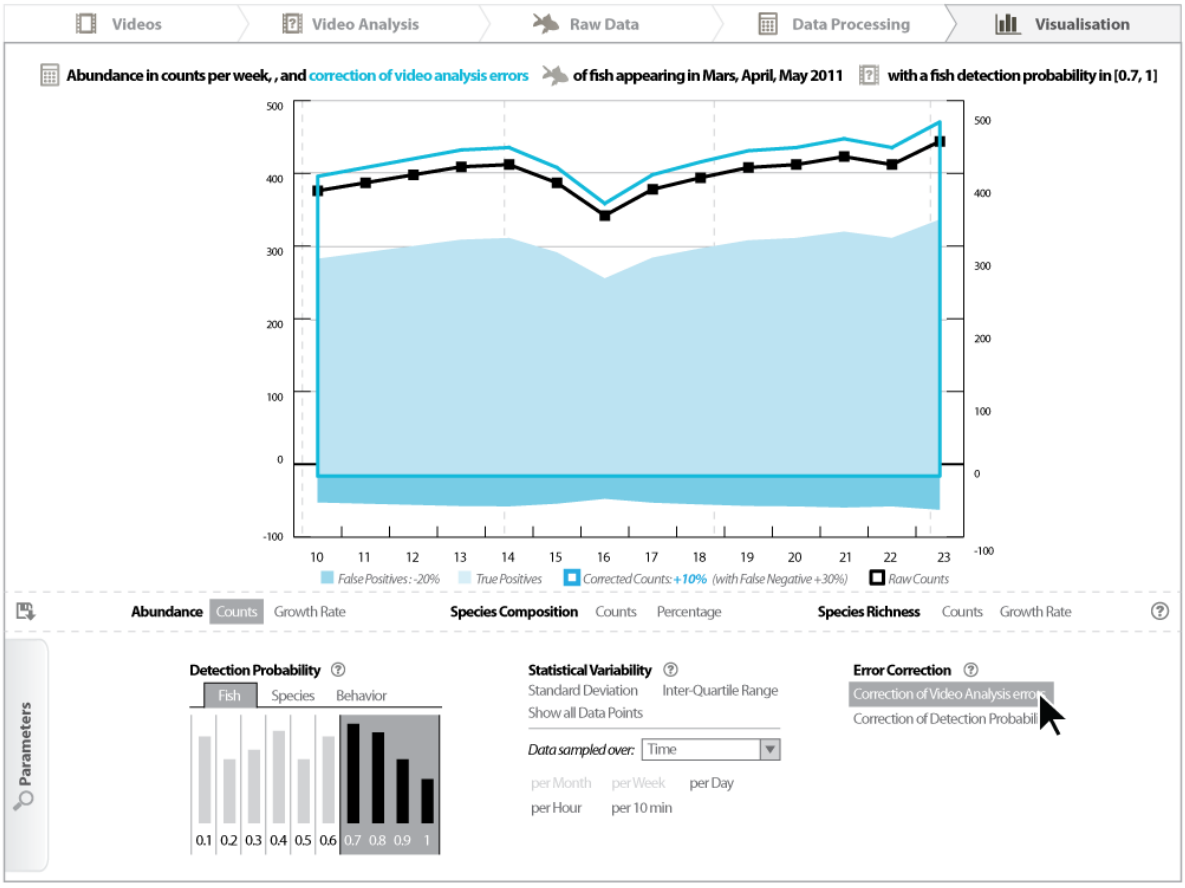


Figure 31: **Estimation of *video analysis errors*.** As described in section 4.1.2, the estimated number of True Positives, False Positives and False Negatives are reported on the population metric results. In the example, the user rolls over the *"Correction of Video Analysis errors"* button (in the bottom right of the UI). This triggers the display of the estimated errors and the corrected count of fish. If the user clicks on the error correction button, this causes the calculation of population metrics using only corrected counts of fish, without the estimated video analysis errors.

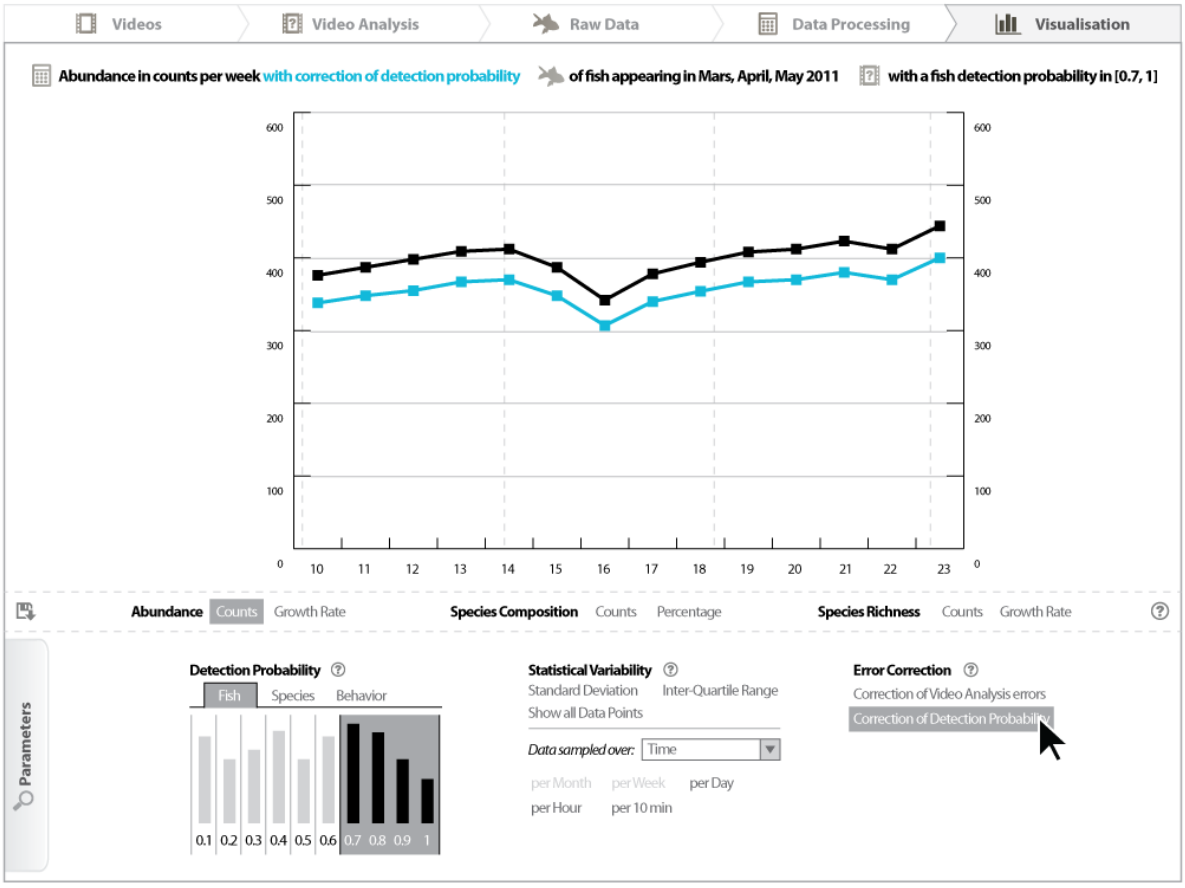


Figure 32: **Estimation of *detection probability errors*.** As described in section 4.1.2, the estimated errors due to imperfect detection probability are reported on the population metric results. In the example, the user rolls over the *"Correction of Detection Probability"* button (in the bottom right of the UI). This triggers the display of corrected count of fish. The counts of fish are corrected using the method defined in section 4.1.2. If the user clicks on the error correction button, this causes the calculation of population metrics using only corrected counts of fish, without the estimated detection probability errors.

5 Evaluation criteria

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).

The trends and correlations of trends in fish populations can be observed in counts of fish which can be calculated depending on 4 variables: the timeframe of fish occurrence, the location of fish occurrence, the species of the fish and the behaviors of the fish. We assume that marine biologists are used to analyzing these counts, and that the identification of trends is a common and well-understood task.

Marine biologists can evaluate the levels of confidence in the identified trends and correlations of trends from the description of the potential errors introduced at each step of the video analysis process. Identifying levels of confidence in the context of automated video analysis is an unusual task in the marine biology domain. It deals with specific errors inherent to image processing, and these errors are different from the errors encountered in more traditional methods for counting fish. Thus the task of identification of levels of confidence (task C above) is the one for which marine biologists need specific support, and the one on which we focus our research effort.

We aim at evaluating the system's ability to support the 3 primary tasks described above, and in particular the identification of levels of confidence, task C. To support them, the interface relies on 2 types of tools we can design and adapt according to user needs: **metrics** (i.e., mathematical tools) that describe the interesting characteristics the data, and **visualizations** (i.e., graphical tools) that conveys these metrics. As mentioned in sections 1.3 and 4.1, we consider the *population metrics* that describe demographics of fish populations, and the *uncertainty metrics* that describe the potential errors inherent to image processing. The population metrics support the identification of trends and correlations of trends in the counts of fish, and the uncertainty metrics support the levels of confidence in the results.

Metrics and visualizations can be separately designed and adapted to users, e.g., we can modify the metrics and keep the same visualization, and vice versa. Thus we aim at distinctively evaluating metrics and visualizations, and we will compare different sets of metrics and visualizations.

The metrics and the visualizations must supply a sufficient amount of information, and that information must be understandable for marine biologists who are not computer vision experts. We will evaluate the UI's ability to supply *sufficient* and *understandable* metrics (i.e., mathematical representations of the fish populations and of the image processing uncertainties) and visualizations (i.e., graphical representations of the metrics) that support the identification of trends, correlation of trends, and users' confidence levels in the results.

The list below gives examples of the types of trends, correlations and levels of confidence we plan to support, and examples of the sets of metrics and visualizations we plan to evaluate.

A. Identification of trends

- Types of trends: increase, stable, decrease.
- Metrics: counts of fish, growth rates, species richness, species composition.
- Visualizations: diagram, line chart.

B. Identification of correlations of trends

- Types of correlations: similar, contrary; and temporality of correlations: precede, simultaneous, follow.
- Metrics: variability of counts over species (i.e., species composition), locations, hours of day, day of lunar month, month of year.
- Visualizations: diagrams of distribution, overlaid line charts, stacked diagram.

C. Identification of level of confidence

- Metrics: False Alarm Rate (FAR), Correct Detection Rate (CDR), False Positives (FP), False Negatives (FN) and True Positives (TP), variability over certainty scores.
- Visualizations: diagrams, scatter plot.
- Confidence levels: very high, high, neutral, low, very low.

The overall goal of the UI evaluation is to analyze how marine biologists understood the metrics and the visualizations, and what characteristics of the mathematical and graphical tools influenced their understanding. In an overall perspective, we aim at studying the semantic gap between the computer vision domain and the marine biology domain, and we aim at drawing conclusions on the implications for the design of mathematical and graphical tools that would support marine biologists. To summarize, we aim at answering the research questions listed below.

- 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?

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.

6 Conclusions and Future Work

We have discussed the uncertainty inherent in the F4K system and their implications on the UI design. On top of that, we sketched our proposed user interface and visualizations that aim at assisting end users in exploiting and understanding the information supplied by the system. Meanwhile, we proposed a set of user tasks and evaluation criteria that will be used for evaluating the proposed UI components.

Our next step is to convert our proposed tasks to one or more concrete experiments using real data from the system and use these to test specific components on users. A number of discussion points are left open for research:

- Are users likely to want to see the ground truth data used? For example to verify the representativeness. E.g., certain species are rare, may not be found in the training set.
- How likely is it that the user wants to combine data sets analysed with different components?
- How do we propagate and combine uncertainties for higher-level operations.
- As our users become more acquainted with the system-introduced uncertainties, they may want to access to more detailed/lower-level information. In other words, what we assume now may change when we start showing the system to users.

References

[1] E. Beauxis-Aussalet and L. Hardman. User scenarios and implementation plan. Technical report, Fish4Knowledge project deliverable D2.2, 2012. URL `http://homepages.inf.ed.ac.uk/rbf/Fish4Knowledge/DELIVERABLES/F4K_Del2-%2_v3-9.pdf`.

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

//homepages.inf.ed.ac.uk/rbf/Fish4Knowledge/DELIVERABLES/Del21.pdf%.

[3] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972 –976, Feb. 2007. doi: 10.1126/science.1136800. URL http://www.sciencemag.org/content/315/5814/972.abstract.

