Fish Detection in Image Sequences

Guilherme Santos
Universidade de Lisboa
Instituto Superior Técnico
Lisboa
guilherme.v.santos@tecnico.ulisboa.pt

The main of this thesis focuses on the detection of regions of interest in images, more specifically the detection of fish in sequences of images. This challenge arose from a need in a study carried out in the hydraulic laboratory of Instituto Superior Técnico, in order to observe the fish trajectories and to characterize their behavior in conditions that simulate the variations of water flow on rivers.

A solution will be presented for the detection of fish in images, more specifically, for the videos obtained in the experiments mentioned above. This solution also includes a section where it is possible to detect the water line representing the water flow in the flume as well as the detection of the fish in the various images and the interconnection of these detections along the video.

The methodologies used as well as the experiments and results will be demonstrated together with the computational simulation process carried out throughout of this dissertation. Finally, the implementation of several algorithms was performed for two datasets obtained in two experiments dating to the spring of 2015 and the fall of 2016.

Keywords— *image features, detection algorithms, regions of interest detection, fish detection*

I. Introduction

The study of marine life is important especially for understanding environmental effects such as pollution, climate changes or symbioses in ecosystems. However, accessing underwater data it is not always easy. Fish behavior analysis is helpful to detect such environmental effects and for tracking change in behavior patterns or finding rare behaviors and detecting the behavioral distinctness of different species. The traditional way of analyzing fish behavior, is usually performed by human visual inspection, that makes this task time consuming and limits the number of processed videos which led to a need to obtain the fish trajectories.

It has been verified that flow discharges downstream in dams, have an impact both on the morphology of the rivers and in the fish community. In rivers subjected to hydropeaking (HP) due to hydropower production the base flow is periodically disrupted by extreme and short-duration fluctuations of discharge during sub-daily peaks of energy demand, raising concerns as to the ability of fish to respond to the quickly changing environment, and the costs and time to react to constant changes [1]. One way of studying the effect of flow variations (peak-flow and base-flow) is to record the behavior of the fish and correlate it to the variation. The main objective of this study is to assess the effects of simulated hydropeaking events on the stress physiology and movement behavior of Iberian barbel (Luciobarbus bocagei) in an experimental indoor flume equipped with lateral shelters. This thesis is related to address a problem posed by the hydraulic department of Civil Engineering of IST. Laboratory Study: the main goal of this research is assessing fish movements and energy expenditure when exposed to different instream structures and rapid flow alterations in an artificial environment (indoor flume, see figure 1), and thus, quantify fish behavior and success of each proposed instream structure. The setup was located at Hydraulics Laboratory (IST – University of Lisbon, Portugal) [1].

Figure 1 - Top view of the original flume setup in R2HP experience made in autumn 2016, the blue arrow indicates the waterflow direction, replotted from [1]

Faced with this flow variation, the fish exhibit a different behavior. One way to evaluate this behavior is precisely by obtaining the recorded trajectories performed by the fishes.

The objective of the thesis is to develop an automatic fish tracking system to enable the evaluation of the fish behavior, which will consist in solving three formulations:

- Waterline detection;
- Fish detection;
- Fish Tracking

In this work, it is proposed an algorithm in image processing capable to achieve the goals above formulated. One potential application of the work will be augmented reality (AR), which could complement the visual information to the operator.

II. State of Art

Active research has been conducted recently on automatic fish recognition, segmentation and counting. In 2013, CSIRO [2] Australia organized a workshop on fisheries and environmental monitoring; a competition of fish recognition was organized by world leading researchers [3] in seeking the solution to fish recognition. Huang et al. [4] (figure 2) proposed a classification tree-based fish segmentation and recognition method to extract fish from complicated background and distinguish their species.

Figure 2 - An example of fish detections from a whole trajectory. This species of fish has a noteworthy tail of white color. This feature is essential to discriminate it from other species of fish. These figures have successfully maintained most of the white tails replotted from [4]

Luo et al. [2] proposed an accurate and automatic algorithm to recognize and count fish in the video footages of fishery operations that it combines machine learning techniques with statistical methods. Lee et al. [5] proposed an intelligent fish recognition system using computer vision to help aquarium to educate people about the fishes in the tank. The designed system provides a convenient and friendly interface to help the users to retrieve the related fish information they are interested. The system includes fish object detection system, fish object tracking system and fish object recognition system. Yao et al. [6] proposed a new fish image segmentation method which is the combination of K-means clustering segmentation algorithm and mathematical morphology.

Researchers attempting to determine animal trajectories from video recordings face the problem of maintaining correct animal identifications after individuals touch, cross or are occluded by environmental features. To bypass this problem, previously, idTracker have implemented the idea of tracking the identification of each individual by using a set of reference images obtained from the video [7] which is a notable attempt to keep the identities correct overtime. They extract characteristic fingerprints from each animal that are matched with the trajectories in the video and use a re-segmentation stage, optimized for shapes that reduces the number of occlusions. idTracker and further developments in identification algorithms for unmarked animals [8] [9] [10] [11] [12] have been successful applied for small groups of 2–15 individuals and in situations with few crossings [11], [13].

Despite such large efforts on fish recognition, the performance of the state-of-the-art multimedia analysis techniques on such task is still far from meeting the real world's requirement in terms of recognize and following fishes. Moreover, recently, deep learning methods have been successfully at object tracking [14] [15] [16] [17] [18], but they are too time consuming for tracking individuals in most datasets, as example.

III. Proposed Solution / Methodology and Overview

A. System overview

In general terms the approach aims to detect and track the fishes in image sequences. This consists in a combination of steps which are a) Waterline detection – for flow estimation and further spatial-constraints b) Fish Detection – which will include the calculation of background image and the method to find the animals, c) Fish Tracking - given a model obtained in the detection step tracks the animal and d) Data - figure out inter frame correspondences between fishes.

Figure 3 represents the sequence of steps used in this approach and the main techniques which will be explained far ahead.

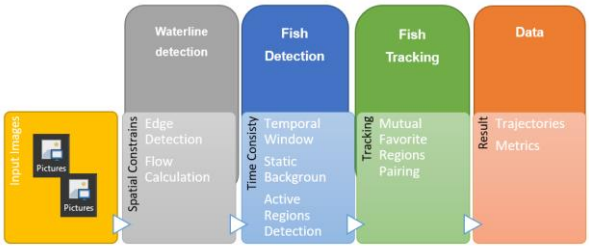

Figure 3 - Proposed solution workflow. The system has three main stages, namely fish detection, fish tracking and data. The first two stages are repeated until all the images of the dataset have been processed.

B. Waterline detection

The waterline algorithm is used to detect in images the value of the yy' coordinates that map the waterline (see figure 4). This spatial constraint will be explained further later. Notice the referential system used in images is with reference in top-left corner being the maximum value of yy' equals to the image height and the maximum value of xx' equals to the image width.

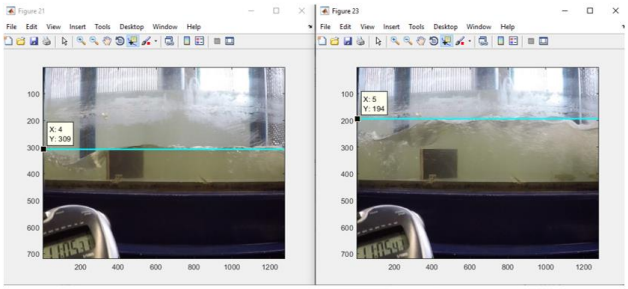

Figure 4 - Waterline detection algorithm showing the yy' coordinates in two different frames

First step of waterline algorithm is the creation of a time window, which is a set of frames defined by:

$$W_i = \{F_t, F_{t+1}, F_{t+n}\} \quad (3.1)$$

The total number of elements of W_i is n, in a video. The spatial constraints will remove most of the outliers and the time validation eliminates the rest. So, at this point for each element of the group W_i is required calculate the Hough transform (HT).

The basic concept of HT is to compute the locus (in the parameter space) of the set of curves passing through each candidate edge element in the given image. An accumulator is assigned to each in which contents are incremented by each locus subtending. The accumulators (with the largest resulting contents) determine the existence (and the coordinates) of the most likely curves in the given image from the specified class. The result of this will be the set of the best line segments for each frame F:

$$L_i \supset \{l_1, \dots, l_n\} \quad (3.2)$$

Where L_i is the set of the best line segments for the frame F_i and l_1 , l_n are the first and last line segments of the set.

For each l_i of the group L_i it is necessary to remove some line segments, to do that its applied:

$$l_i: (\alpha \cdot l_i) > \left(\frac{\pi}{2} + \xi\right) \quad (3.3)$$

where α is the angle formed between the line segment and the reference frame xx and ξ a threshold. Now we have L_α which contains only the horizontal lines (which may be water lines). Defining a RANSAC (Random sample consensus) threshold which will regulate the number of contributions needed to obtain the water line.

Using RANSAC for waterline detection will work this way:

- Draw s points uniformly at random (points which belong to lines from L_α)
- Fit line to these s points - Hypothesize a model
- Find inliers to this line among the remaining points (i.e., points whose distance from the line is less than t - Compute error function)
- If there are inliers, accept the line and refit using all inliers - Select points consistent with model
- Repeat hypothesize and verify loop

After repeating this process for the remaining lines of the set L_α and comparing the value of the yy' coordinate in each line we can conclude if the lines have yy' coordinates value similar they are static then it is due to the movement of the water, as such is the limit for the water line in the aquarium whereas if the line changes its coordinate yy' they don't belong to the water line.

C. Fish Detection

Ideally, the detection block should detect all the fishes in the image, keeping the number of false alarms as small as possible. The first algorithm is denoted as basic background subtraction (BBS) algorithm. It computes the absolute difference between the current image and a static background image and compares each pixel to a threshold. All the connected components are computed, and they are considered as active regions if their area exceeds a given threshold. In BBS [19] algorithm the regions of interest are detected when comparing the reference frame with the background image of the set. A difference is computed in each pixel and a threshold operation is performed to classify each pixel as foreground or background. If

$$\|I^t(x,y) - \mu^t(x,y)\| > T \quad (3.4)$$

where $I^t(x,y)$ is a 3x1 vector being the intensity of pixel (x,y) in the current frame t, $\mu^t(x,y)$ is the mean intensity (background) of the pixel, and T is a constant user defined which is the threshold.

Ideally, pixels associated with the same object should have the same characteristics. This can be accomplished by performing a connected component analysis. This step is usually performed after a morphological filtering or spatial constraint to eliminate isolated pixels and small regions

The second method follows is a Single Gaussian Model (SGM) assumes that each pixel of the background is formed by a random variable with Gaussian distribution [20], the mean and covariance of the Gaussian distribution are independently estimated for each pixel. In this approach, the information is saved in a vector $[Y,U,V]^T$, which defines the color and intensity values for each pixel in image. To calculate the background scene each pixel needs to be recursively update as so the mean $\mu\,(x,y)$ and covariance $\sum(x,y)$ values associated to each pixel. The update is as follows:

$$\mu^t(x,y) = (1-\alpha)\mu^{t-1}(x,y) + \alpha I^t(x,y) \quad (3.5)$$

$$\sum^t(x,y) = (1-\alpha)\sum^{t-1}(x,y) + \alpha\big(I^t(x,y) - \mu^t(x,y)\big)(I^t(x,y) - \mu^t(x,y))^T \quad (3.6)$$

where $I\,(x,y)$ is the pixel of the current frame in $[Y,U,V]$ color space and α is a constant. The algorithm performs a binary classification of the pixels into foreground or background and tries to cluster foreground pixels into blobs. Pixels in reference frame are compared with the background by measuring the log likelihood in color space. This way it is possible to categorize pixels as belonging to the background or foreground region. To classify as an active pixel, need to have a small likelihood, that is computed using the following expression:

$$L\,(x,y) = \frac{1}{2}\,(I^t(x,y) - \mu^t(x,y))^T(\textstyle\sum^{-1})^T\,\big(I^t(x,y) - \mu^t(x,y)\big) - \frac{1}{2}\ln|\textstyle\sum^t| - \frac{m}{2}\ln(2\pi) \quad (3.7)$$

where $I^t(x,y)$ is a vector $[Y,U,V]^T$defined for each pixel in the reference image, $\mu^t(x,y)$ is the pixel vector in the background image, Bkg. If the value of the likelihood was exceeded, then the pixel belongs to the background.

The third method represents the distribution of the background pixels as a Mixture of Gaussian Model (MGM) [21], displaying each pixel $I\,(x,y)$ as a mixture of N (N=3) Gaussians distributions i.e.,

$$p\big(I\,(x,y)\big) = \textstyle\sum_{k=1}^{N} w_k\, N(I\,(x,y), \mu_k(x,y), \sum_k(x,y)) \quad (3.8)$$

where $N(I\,(x,y), \mu_k(x,y), \sum_k(x,y))$ is a multivariate normal distribution and w_k is the weight of kth normal,

$$N(I\,(x,y), \mu_k(x,y), \textstyle\sum_k(x,y)) = c\, e^{\left\{-\frac{1}{2}(I\,(x,y) - \mu_k(x,y))^T \sum_k^{-1}(x,y)(I\,(x,y) - \mu_k(x,y))\right\}} \quad (3.9)$$

with $c = \frac{1}{\left(2\pi^{\frac{n}{2}}|\sum_k|^{\frac{1}{2}}\right)}$. Note that each pixel $I(x,y)$ is a 3x1 vector with three component colors (red, green and blue) i.e., $I(x,y) = [I(x,y)^R\, I(x,y)^G\, I(x,y)^B]^T$. To avoid excessive computational cost, the covariance matrix is assumed to be diagonal. The mixture model is dynamically updated, each pixel is updated as follows routine. First the algorithm checks if each incoming pixel value x can be ascribed to a given mode of mixture. Secondly if the pixel value occurs inside the confidence interval with $\pm$ 2.5 standard deviation, a match event is verified. The parameters of the corresponding distributions for that pixel are updated according to

$$\mu_k^t(x,y) = (1-\lambda_k^t)\,\mu_k^{t-1}(x,y) + \lambda_k^t I^t(x,y) \quad (3.10)$$

$$\textstyle\sum_k^t\,(x,y) = (1-\lambda_k^t)\sum_k^{t-1}\,(x,y) + \lambda_k^t\big(I^t(x,y) - \mu_k^t(x,y)\big)\big(I^t(x,y) - \mu_k^t(x,y)\big)^T \quad (3.11)$$

$$\text{where } \lambda_k^t = \alpha\, N(I^t(x,y), \mu_k^{t-1}(x,y), \textstyle\sum_k^{t-1}(x,y)) \quad (3.12)$$

The weights are updated by:

$$w_k^t = (1-\alpha)w_k^{t-1} + \alpha(M_k^t) \quad (3.13)$$

$$\text{with } M_k^t = \begin{cases} 1, matched\ models \\ 0, remaining\ models \end{cases} \quad (3.14)$$

α is the learning rate. The non-match components of the mixture are not modified. If none of the existing components

match the pixel value, the least probable distribution is replaced by a normal distribution with mean equal to the current value, a large covariance and small weight. The next step is a need to order the distributions in the descending order of w/σ, to favors the distribution which have more weight and less variance. Finally, the algorithm models each pixel as the sum of the corresponding updated distributions. The first gaussian modes are used to represent the background, while the remaining modes are considered as foreground distributions. Bkg is chosen as follows: Bkg is the smallest integer such that:

$$\sum_{k=1}^{Bkg} w_k > T \quad (3.15)$$

where T is a threshold that accounts for a certain quantity of data that should belong to the background.

The BBS, SGM and MGM algorithms use color images for its implementation, it is necessary to determine the image background as referred to in the previously and then the next step is to calculate the active regions in the image.

The three algorithms described above allow obtaining the active regions where the image movement occurs, many of them could be false alarms, that is, active regions that do not contain fishes. A noise removal step is necessary, for removing regions that do not have an area similar to the object (fish) to achieve this, we will impose spatial constraints to the detection algorithm.

We need further processing to separate regions of interest from other misunderstandings. In order to remove the false alarms, is used a spatial constraint methodology in which all the regions detected with yy' coordinates value lower than the value of yy' coordinates of the water line in the same frame will be rejected (figure 5-bottom).

The second criteria will be all the regions detected near the borders will be deleted, the procedure is similar with the described above, the term for compare is if the coordinates of the region detected is near to the borders of the image, yy' and xx' maximum and minimum.

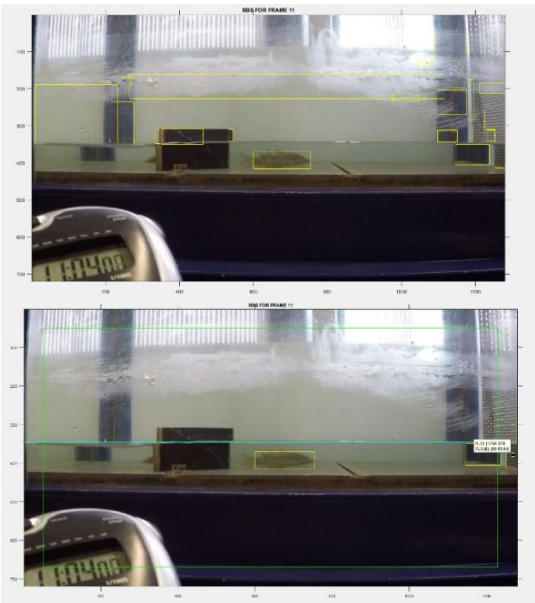

Figure 5 - Spatial Constraints: (Top) – original frame of Data1 dataset from autumn 2016 R2HP experience; (Bottom) – output with only 2 regions detected: fishes (at right) false detection or false positive (at left).

D. Fish Tracking

Finally, it is necessary to map the same active regions over time to obtain trajectories. This tiling can be done by:

- Using the algorithm of Mutual Favorite Pairing (MFP). Here two regions are matched, if the overlap is significant. Only spatial information is used.

Object matching is performed by computing a binary correspondence matrix $C(a,b)$, which defines the correspondence between the active regions on a pair of images. Let us assume that we have N detected regions in one frame $\widetilde{R_a}$ and M detected regions in the subsequent frame $\widetilde{R_b}$. Under this conditions $C(a,b)$ is a $N \times M$ matrix, defined as follows:

$$C(a,b) = \begin{cases} 1 \; if \; \frac{\widetilde{R_a} \cap \widetilde{R_b}}{\widetilde{R_a} \cap \widetilde{R_b}} > T \\ 0 \; if \; \frac{\widetilde{R_a} \cap \widetilde{R_b}}{\widetilde{R_a} \cap \widetilde{R_b}} < T \end{cases} \quad \forall a \in \{1, \dots, N\}, b \in \{1, \dots, M\} \quad (3.16)$$

where T is the threshold which accounts for the overlap requirement.

The region-based measures described here depends on an overlap requirement T between the detected regions of two consecutive frames. Without this requirement, this means that a single pixel overlap is enough for establishing a match between a detected region in reference frame and the detected region in the posterior, which does not make sense. A match is determined to occur if the overlap is at least as big as T. The bigger the overlap requirement, the more the pixels are required to overlap.

IV. Experimental Implementation

A. Experimental Setup

The main objective of this study is to assess the effects of simulated hydropeaking events on the stress physiology and movement behavior of Iberian barbel (Luciobarbus bocagei) in an experimental indoor flume equipped with lateral shelters. Specifically, the following null hypothesis were tested: 1) flow magnitude and hydropeaking event duration do not cause significant changes in the levels of blood glucose and lactate of Iberian barbels and on the movement behavior of Iberian barbel in the flume; 2) Iberian barbels use equally the available shelters when subjected to peak-flows and base-flows.

The indoor flume, which will represent the living environment of the fishes has a rectangular cross-section 8.0m long, 0.7m wide and 0.8m high, with a steel frame and glass viewing panels on both sides, and with flap gates installed up and downstream the flume allowing to create rapid flow changes, as showed above in figure 6, an upstream reservoir which was fed by the main laboratory reservoir through a hydraulic network, and a downstream recovery compartment that directed the water to the main laboratory reservoir. The flume is equipped with a PVC false bottom that will be used to fix the proposed instream structures (e.g. deflectors). In each experiment fish will be exposed to rapid variations of flow and different metrics (e.g. use of shelter, sprint, drifts, jumps) will be accounted by visual observation

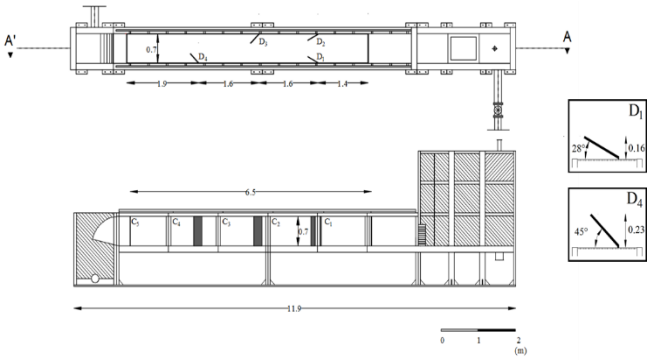

Figure 6 - Top (up) and side (bottom) view of the indoor experimental flume with representation of the artificial shelters (deflectors: D1 to D4) and behavior observation compartments (C1 to C5). D1 and D4 represent the detailed dimensions of these two deflectors

The HP conditions were simulated up to 60ls^{-1} discharge, which is controlled by an upstream plane gate allowing rapid variations in flow. The usable area for fish is limited by two perforated metallic panels creating a 6.5 m reach usable for fish. To mimic lateral shelters in a river channel, termed deflectors as D1 – D4 were installed in the false bottom of the flume. With this configuration, the flume water velocities were maximized, creating a harsher hydraulic environment for fish. The deflectors were installed in a configuration characteristic of meandering river reach, creating a more heterogeneous flow environment. Several shelter configurations were used on this experiment, figure 7 shows more information about it.

Shelter type	Shelter dimensions (cm)	Shelter material
deflector	l=30 h=76 w=1,5	PVC
small pyramid	pyramid h=6.3 a= 20 h=10.2	Plywood
big pyramid	pyramid h=10.5 a=30.5 h=15.5	Plywood
open triangle	pyramid h=10.5 a=30.5 h=16.8	Plywood

Figure 7 - Shelter configuration variations

The position of the deflectors on the flume are represented on figure 8:

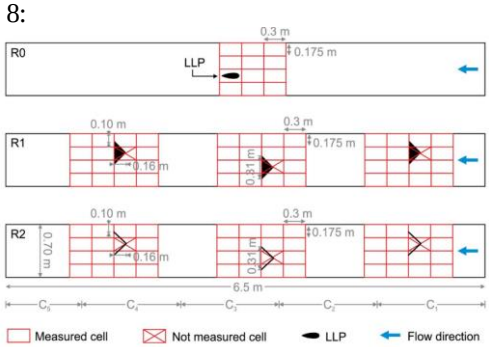

Figure 8 - Setup of the flume with dimensions and shelter representation, C1 to C5 indicates the compartment, replotted from [1], the triangles represent shelter positions on compartments, the blue arrow indicate the flow direction and R's determine the type of shelter used, R0 – no shelter, R1 – small pyramids and R2 – open triangles, respectively.

B. *Dataset*

This section presents the configuration of the experiments which will serve as dataset inputs for the algorithms. Figure 9 shows the important details to have in count.

Dataset Name	Original Name	Number of Frames	Resolution (pixels)	Shelters	Details
Data1	Spring 2015 R0HP	300 – C1	720x1280	No	C1 – compartment 1
Data2	Autumn 2016 R2HP	2482 – C1 2482 – C2_3 2482 – C4_5	720x1280	Yes – open triangles	C1 – compartment 1 C2_3- compartment 2 and 3 C4_5 – compartment 4 and 5

Figure 9 - Datasets

The figures bellow (figure 10 and 11) represents the Data1 and Data2 cameras configuration as also the position and type of deflectors used.

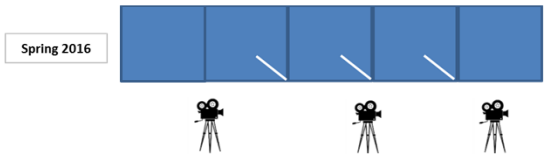

Figure 10 - Data1 (Spring 2015 R0HP) flume configuration, each blue square represents a compartment, on compartment 1 there was no shelter installed.

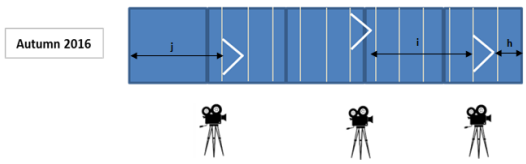

Figure 11 - Data2 (Autumn 2016 R2HP) flume configuration, each blue square represents a compartment, h=87cm, i=183cm and j=147cm

Figure 12 shows the HP event:

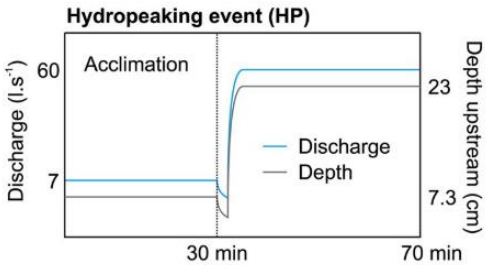

Figure - 12 - Depth and discharge timing in experiences made related with the dataset, replotted from [1]

The mean fish size is total length ± standard deviation: 18.0 ± 3.8 cm; mean total weight ± standard deviation: 56.3 ± 36.6 g.

Each treatment comprised a group of five Iberian barbels and their frequency will be registered for each individual and group of 2 to 5 individuals in each compartment. A trajectory was only considered if it started in that compartment and it will be characterized by compartment, this means that if the individual moves to another compartment the trajectory ends.

C. Evaluation Metrics

Matching strategy and determining which feature matches are reasonable to process further depends on the context in which the matching is being performed. We can quantify the performance of a matching algorithm at a particular threshold by first counting the number of true and false matches and match failures, using the following definitions[22]:

- True positives (TP), i.e., number of correct matches;
- False negatives (FN), matches that were not correctly detected;
- False positives (FP), proposed matches that are incorrect;
- True negatives (TN), non-matches that were correctly rejected.

The evaluation of the performance of object detection algorithms in video sequences was already studied [23].To evaluate the performance of object detection algorithms it's proposed a framework which is based on:

- A set sequence is selected for testing and all the moving objects are detected using automatic procedure and manually corrected if necessary to obtain the ground truth.
- The output of the automatic detector is compared with the ground truth.
- The errors are detected and classified in classes as explained before.
- For each detection a set of statistics (trajectory, velocities) are computed.

To perform this will be made a user interface which allows user to define the foreground regions in test sequence in a semi-automatic way. A set of frames is extracted from the test sequence. An automatic object detection algorithm is then used to provide a tentative segmentation of the test images. Finally, the automatic segmentation is corrected by the user, by merging splitting, removing or creating active regions, figure 13 shows an example of this user interface where the user is correcting the fish detection in shelters.

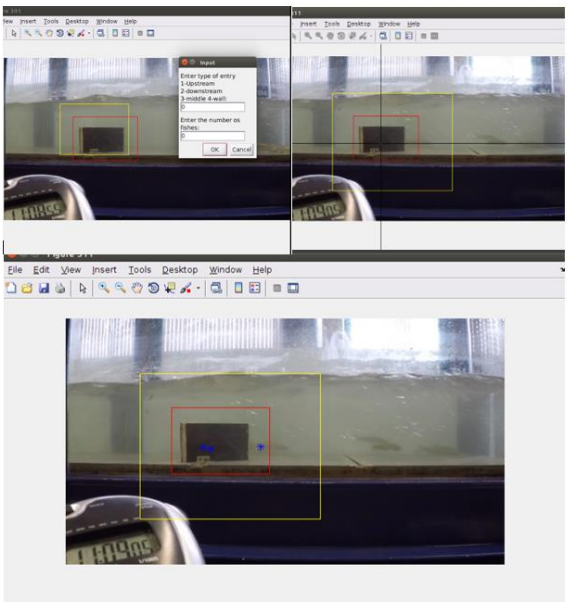

Figure 13 - User interface for correct fish detections in shelters

V. Simulations, Results and Discussion

The detection of the waterline in picture is important because it will allow to calculate the flow of the water and relate it with metrics available to distinguish the velocity of water and consequently the speed at which the fish are subjected.

To help the algorithm a user defined band is selected in three frames of the data set limiting the wide area of search. The following images (figure 14 and 15) contains the line drawn manually by the user (with red color), it attends as a reference for ground truth and the algorithm detected line (with cyan color), for two frames of the datasets: Data1 and Data2.

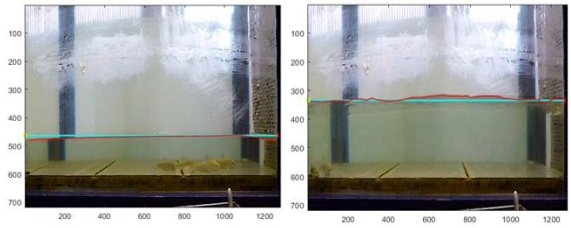

Figure 14 - Waterline detection algorithm after RANSAC step applied in Data1 dataset, the red line is manually detected by the user and the cyan line is automatic detected by the algorithm which represent the waterline in flume

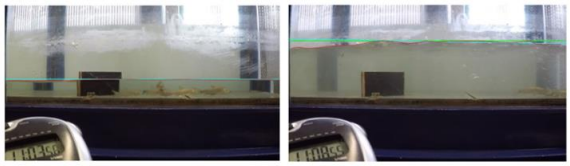

Figure 15 - Waterline detection algorithm as showed in figure 23 but applied on Data2 dataset, the green line represents other solution (not the best) when RANSAC is applied

The next image (figure 16) will show the hydropeaking event, it is possible to observe a step, representing the rise of water in flume

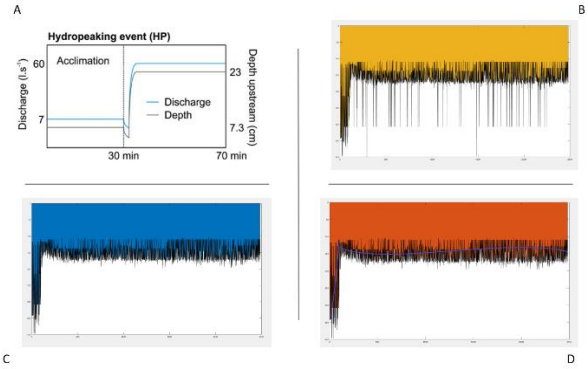

Figure 16 - Graphs that represents the yy' coordinate of the waterline detection along the Data2 dataset: A: Hydropeaking event registered at the experiences; B: waterline detection for [1:2482] frames of the dataset, representing the x axis, y axis in pixels [1:400]; C: waterline detection as B with noise removal; D: same as C with a trendline similar as the HP represented in A

As can be seen from the figure there are, in some frames, some erroneous waterline detections, these outliers were removed manually, the rise-up of the water is until frame 135, is equivalent to 2:25 min of the video and consistent with autumn R2HP experiments data.

In order for this detection to be valid it is necessary to calculate the background of the data set, therefore the training of the background for the BBS, SGM and MGM (figure 17) algorithms was made with 150 frames selected in the middle of the dataset and the result if the following:

Figure 17 - Background computation with MGM for Data2 dataset

Hereafter background prediction it is now possible to start to detect the active regions so that, after having successfully detected the fish to be now possible to create a tag for each so that its trajectory can be followed along the video. Initially, only 2 algorithms were used to make the detection: BBS and SGM. In the following images we can see the result.

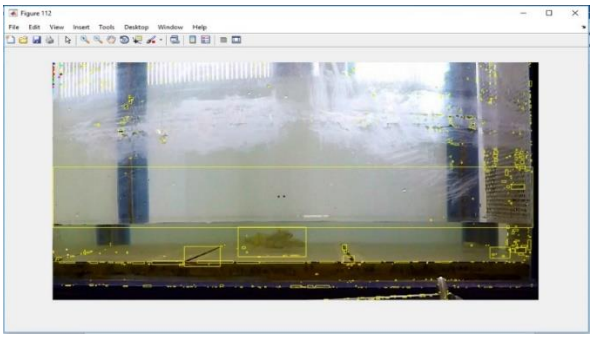

Figure 18 - BBS active region detection output for Data1

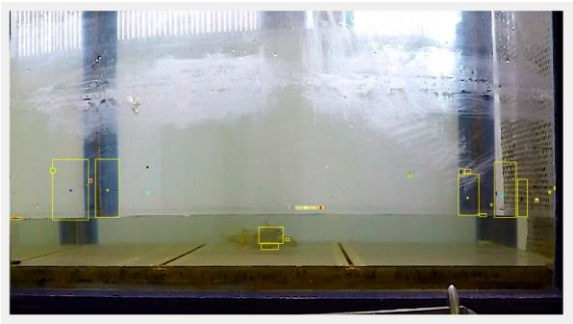

Figure 19 - SGM active region detection output for Data2

VI. Conclusions and Future perspectives

Current animal multitracking systems calculate the most likely assignment of identities by considering the movement of the animals before and after an overlap. Some of these systems incorporate image processing techniques to separate the images of the individuals when the overlaps are small or use specific shape models, normally adapted to the species that can help to resolve more complex crossings. The first problem was how can be preserve the identity of each animal when tested a group of five individuals each time.

In general, there is no clear consensus on which line estimation technique performs best. It is therefore a good idea to think carefully about the problem at hand and to implement several approaches to determine the one that works best for your application.

Looking at the literature regarding this topic there are still challenges ahead. The use of various approaches describing the evolving scene at pixel level behavior of a continuous movement was the step needed in background modeling and subtraction.

In the use of RANSAC, an advantage is that no vector accumulator is required and therefore the algorithm may be more efficient. The disadvantage is that many other assumptions may need to be generated and tested than those obtained by finding peaks.

Sometimes the segmentation procedure is subjective, since each active region may contain several objects and it is not always easy to determine if it is a single connected region or several disjoint regions. The spatial constraints will\ remove most of the outliers and the time validation eliminates the rest. We conclude that a reliable detection of the fishes is therefore possible. Most of the current approaches to segmentation depend on image-based criteria, such as color, grey level, or texture uniformity of image regions; the smoothness and continuity of their bounding contours or a combination of these. The region-based approaches merge and split image regions according to specific criteria. Merging approaches recursively merge similar regions. Contour-based approaches emphasize the properties of region boundaries, such as continuity, smoothness, length, curvature, and shape.

Image Segmentation for object detection is the most vital part of computer vision where the computer has to identify objects differently from the background whether it is a face or hand or a man or simply static objects. If the contrast difference of the background with the foreground is high, then the detection is simple. But if the background is cluttered and there is a little difference between the background and the object then it becomes difficult for the system to identify the edges from the background. The major issues in object detection are the shape variance, lighting variance and objects.

In future works would be nice to have an extra module on proposed model to do tracklets relinking which should be a postprocessing stage which further solves trajectory fragmentation caused by occlusion and detection error.

In object tracking with recourse to the use of optical flow. In this method, the Lukas-Kanade [15] algorithm is used. In this method, the optical flux in the area of the image delimited by the active region is calculated. That is, temporal information is used. The next step will be to calculate the velocity histogram (each bin of the histogram is associated with a given velocity). The pairing of two regions will be performed if the intersection or similarity of their histograms is significant.

The implementation of augmented reality will provide to the user a better understanding view of the results and complement the visual information of metrics. This information can include jumps, drags, sprints and even the shelters used by the fish. This AR will be able to deliver high rates of transmission to users.

VII. References

[1] M. J. Costa, J. F. Fuentes-Pérez, I. Boavida, J. A. Tuhtan, and A. N. Pinheiro, “Fish under pressure: Examining behavioural responses of Iberian barbel under simulated

hydropeaking with instream structures," *PLoS One*, vol. 14, no. 1, p. e0211115, Jan. 2019.

[2] S. Luo, X. Li, D. Wang, J. Li, and C. Sun, "Automatic Fish Recognition and Counting in Video Footage of Fishery Operations," in *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, 2015, pp. 296–299.

[3] "http://www.lifeclef.org, Nov 2017," 2017. [Online]. Available: http://www.lifeclef.org.

[4] P. X. Huang, B. J. Boom, and R. B. Fisher, "Hierarchical classification with reject option for live fish recognition," *Mach. Vis. Appl.*, vol. 26, no. 1, pp. 89–102, Jan. 2015.

[5] J.-H. Lee, M.-Y. Wu, and Z.-C. Guo, "A tank fish recognition and tracking system using computer vision techniques," in *2010 3rd International Conference on Computer Science and Information Technology*, 2010, vol. 4, pp. 528–532.

[6] H. Yao, Q. Duan, D. Li, and J. Wang, "An improved K-means clustering algorithm for fish image segmentation," *Math. Comput. Model.*, vol. 58, no. 3–4, pp. 790–798, Aug. 2013.

[7] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. De Polavieja, "IdTracker: Tracking individuals in a group by automatic identification of unmarked animals," *Nat. Methods*, vol. 11, no. 7, pp. 743–748, 2014.

[8] R. Dolado, E. Gimeno, F. S. Beltran, V. Quera, and J. F. Pertusa, "A method for resolving occlusions when multitracking individuals in a shoal," *Behav. Res. Methods*, vol. 47, no. 4, pp. 1032–1043, Dec. 2015.

[9] M. J. Rasch, A. Shi, and Z. Ji, "Closing the loop: tracking and perturbing behaviour of individuals in a group in real-time," *bioRxiv*, p. 071308, 2016.

[10] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, and M. Andersson, "ToxId: An efficient algorithm to solve occlusions when tracking multiple animals," *Sci. Rep.*, vol. 7, no. 1, pp. 1–8, 2017.

[11] S. H. Wang, J. W. Zhao, and Y. Q. Chen, "Robust tracking of fish schools using CNN for head identification," *Multimed. Tools Appl.*, vol. 76, no. 22, pp. 23679–23697, Nov. 2017.

[12] Z. XU and X. E. Cheng, "Zebrafish tracking using convolutional neural networks," *Sci. Rep.*, vol. 7, no. 1, p. 42815, Dec. 2017.

[13] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. H. Heras, and G. G. de Polavieja, "idtracker.ai: tracking all individuals in small or large collectives of unmarked animals," *Nat. Methods*, Jan. 2019.

[14] H. Nam and B. Han, "Learning Multi-Domain Convolutional Neural Networks for Visual Tracking," 2016.

[15] M. Zhai, L. Chen, G. Mori, and M. J. Roshtkhari, "Deep learning of appearance models for online object tracking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 11132 LNCS, no. July, pp. 681–686.

[16] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video Tracking Using Learned Hierarchical Features," *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1424–1435, 2015.

[17] H. Nam, M. Baek, and B. Han, "Modeling and Propagating CNNs in a Tree Structure for Visual Tracking," pp. 1–10, Aug. 2016.

[18] R. T. Collins *et al.*, "A System for Video Surveillance and Monitoring: Vsam Final Report Robotics Institute," Pittsburgh, PA, 2000.

[19] R. C. Gonzalez and R. E. Woods, *Digital Image Processing, 2nd ed.*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[20] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[21] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, 2000.

[22] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[23] J. C. Nascimento and J. Marques, "Performance evaluation of object detection algorithms for video surveillance," *IEEE Trans. Multimed.*, vol. 8, no. 4, pp. 761–774, Aug. 2006.