Fish tracking in river flows

Catarin Freire Pimentel Pessoa Jorge[1]

Abstract— **This work explores the impact of flow discharges downstream of dams on fish behaviour. These discharges occur more and more frequently, with sensed impacts on the morphology of the rivers and the biological communities that live there. It refers particularly to the fish community, the focus of this study. In this way, the development of a methodology capable of automatically monitoring fish behaviour is proposed, using image processing and analysis.**

For the analysis of the behaviour of the fish, a video provided by the department of hydraulics of Instituto Superior Técnico (IST) is used. With the aid of this video, a solution is proposed that is able to identify the location of the fish as well as the river flow line.

For the identification of the river flow line, two strategies are studied. The first is the use of a region of interest that is dynamic along the sequence of images (strip). The second strategy is to use dynamic programming. For the identification of the fish, two object detection algorithms were considered: (i) Basic Background Subtraction (BBS) and (ii) Single Gaussian Model (SGM). Both algorithms are capable of separating the background from the active regions that occur in the sequence (foreground). After the above-mentioned identification, we proceed to study how the locations of the active regions evolve over time. In order to do so, we can pair these regions using algorithms (Mutual Favorite Pairing) capable of providing temporal information.

As results, it has been found that fish have a higher mean velocity during water variation and consequent disturbance in the profile line/flow boundary.

I. INTRODUCTION

Downstream flow discharges into dams are a factor that has a direct impact on both river morphology and the fish community. In order to analyze this impact, the hydraulics department of IST assembled and filmed an experimental channel in which the water flow was varied. In this way it is possible to develop a methodology capable of automatically monitoring fish. This automatic monitoring allows to acquire data of the movements of fish and therefore to study their behavior, being this data important for biological and robotic research.

The aim is therefore to develop a solution that meets existing needs, using new technologies and developing new methodologies to support biological research [1] [2].

II. RELATED WORK

The detection and monitoring of fish has been an area of investigation with the objective of conducting a behavioural analysis of the fish when the natural conditions of their habitat changes. One way to achieve this goal is to use image processing and analysis techniques. The first stage of a solution is to detect the positions/locations of the fish (i.e. active regions) at a given time. This information is crucial for the later calculation of trajectories, i.e. detection of regions active over time.

There are many works related to this subject, such as the works of Cigdem Beyan and Robert B. Fisher, who began by proposing in 2012 [3] a proposal that aims to identify unusual behaviour in fish in order to understand the effect that the environmental impact has on these animals. This study presents a rule-based trajectory filtering mechanism to extract normal fish trajectories and potentially helps to increase the accuracy of abnormal behaviour detection systems and can be used as a preliminary method. As early as 2013 [4], the same researchers published another paper proposing a solution to analyse fish trajectories. They consider that there are two types of trajectories: the first consists of the 'normal' trajectory, which consists of the usual behaviour of the fish and the 'abnormal', that corresponds to the unusual behaviour of the fish. This solution is based on the use of a hierarchical classifier that groups data based on their similarity whereas different sets of characteristics are distributed by different hierarchical levels. This method of analysing fish trajectories has had an appreciable performance when compared to other more innovative methodologies, thus bringing significant results for this area.

The work [5] proposes a solution that allows estimating the movement of the fish in real time based on adaptive appearance models and strategies of tracking, that can be adapted to several changes of the appearance of the fish caused by not rigid deformation. The solution is to use a model with two fish appearances that is adapted during tracking. Initially the algorithm Kernelized correlation filter (KCF) [6] is used to do the tracking of the fish, since this is one of the best in terms of speed and precision compared to other works. This method works as follows:

> In the first frame, the algorithm is trained by surrounding the target and storing the target in a memory (array) structure. Subsequently, in the test phase it is performed based on the previous position of the target, ie, the same set of points (target) is sought in the vicinity of the previous positions. After the identification of the target, it is represented in a bounding box, thus achieving the location of the active region.

However this model alone is not able to deal with fish deformations, as such a model was constructed capable of dealing with the multiple appearance (normal and abnormal). The normal appearance is when the fish is in linear motion while the abnormal mode is when it is in spinning mode.

[1]Instituto Superior Técnico

Through this model it is possible to have a greater robustness in the real time tracking of fish according to the results presented in the article.

The work [7] proposes a solution for the video processing system, dividing it into three subsystems: video texture analysis, fish detection and module tracking. The detection of the fish is made from the result of two independent algorithms, the Single Gaussian Model, and the H-Model, which together achieve greater accuracy . Tracking is accomplished using the *CamShift* [8] algorithm that allows tracking of objects whose numbers can vary over time. For fish counting, they propose an approach that allows these animals to be counted in all types of environments and under different scenarios (dark water, algae in the camera lens, moving plants, low contrast, etc.).

III. PROPOSED METHODOLOGY

A. Pre-processing

This phase explains the pre-processing performed on the video in order to facilitate the implementation of the solution. Initially we started by eliminating the first two seconds of the video because they are unnecessary to the training phase (signalling given by the user to start the recording). The bottom part of the frames of the video was also removed so that the aquarium was visible.

After these changes it was necessary to extract all the frames of the video, having a total of 18137 frames in the end. These frames at the end also suffered a reduction in size from 1280x400 pixels to 768x240 pixels, in order to reduce the processing time of the algorithm.

As the processing time of 18137 frames was still very high, the video was divided into 3 distinct parts. This division is detailed in the table:

TABLE I
SPLITTING THE VIDEO SEQUENCE

	First sequence	**Second sequence**	**Third sequence**
Frames	901 - 1500	1501 - 2700	8501 - 9200
Total frames	599	1199	699

These frames were chosen taking into account the most important sequences of the video, that is, where the fish appear more regularly, as well as the part where the greater variation of the water line takes place.

This division also had an impact on fish tracking, since each of the sequences has a distinct test phase, this being important because the background is different in each of the sequences, in view of the rise in river flow. As such, obtaining different training steps for different video sequences allowed the testing phase to be more efficient in obtaining the active regions, i.e. fish.

B. Fish Tracking

1) Fish detection in the image: In order to proceed to the detection of the active regions, two algorithms were considered, Basic Background Subtraction (BBS) and Single Gaussian Model (SGM). These algorithms were both studied to verify which one provides the best performance in the detection of fish.

- **Basic Background Subtraction (BBS)** is an algorithm that consists of separating the background from the foreground.
  This algorithm has two phases, one of training and one of testing. The training phase consists of obtaining a background image which is the result of the application of a low-pass filter over time. This filter intends to attenuate the high spatial frequencies of the image, that is, it allows the removal of noise in the image as well as the objects that move in the scene. This allows the background to have only the content constant in the scene. The integrator is based on the following equation:

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

  Where $\mu(x,y)$ is the mean pixel intensity vector, $I(x,y)$ represents a vector with the pixel intensity in the current image and α is an adjustable parameter (α between 1 and 0).
  In the test phase the absolute difference between the background image and the current image is made, comparing each pixel with a threshold. If this difference exceeds the defined threshold, then this pixel will correspond to an active region, that is, it will represent a moving object. This operation can be described by the equation:

$$|I(x,y) - \mu(x,y)| > T \quad (2)$$

  Where $I(x,y)$ represents a vector with the intensity of the pixel in the current image, $\mu(x,y)$ is the mean of the pixel intensity of background learned in the training phase and T is a constant.

- **Single Gaussian Model (SGM)** assumes that each pixel of the background corresponds to the realization of a random variable with the Gaussian distribution [9], with the objective of collecting the color of each pixel to a vector in the YUV color space.
  This algorithm, like the BBS, has the training phase and the test phase. In the training phase, the mean $\mu(x,y)$ and the covariance $\Sigma(x,y)$ of the Gaussian distribution are independently estimated for each pixel by means of the expressions of the equations 3 and 4, respectively. This information is updated for each frame of the video sequence.

$$\mu^t(x,y) = (1-\alpha)\mu^{t-1}(x,y) + \alpha I^t(x,y) \quad (3)$$

$$\Sigma^t(x,y) = (1-\alpha)\Sigma^{t-1}(x,y) + \alpha(I^t(x,y) - \mu^t(x,y)) \times (I^t(x,y) - \mu^t(x,y))^T \quad (4)$$

  Where $I^t(x,y)$ corresponds to the pixel in the current frame in the YUV color space and α is a constant to be adjusted.

The likelihood function (equation 5) of each pixel is calculated. More specifically, the 1st and 2nd order statistics of the Gaussian probability density function (i.e., media and covariance) are calculated. The likelihood function is obtained by the Mahalanobis distance as the following equation describes:

$$l(x,y) = -\frac{1}{2}(I^t(x,y) - \mu(x,y))^T(\Sigma^{-1})(x,y)\times \\ (I^t(x,y) - \mu(x,y)) - \frac{1}{2}ln|\Sigma| - \frac{m}{2}ln(2\pi) \quad (5)$$

$I^t(x,y)$ is a vector $(Y,U,V)^T$ defined for each pixel in the current frame and $\mu(x,y)$ is the average pixel of the estimated background.

In the test phase, the likelihood function is compared with a given threshold. If the pixel likelihood function is above a certain threshold, the pixel is considered to be active.

If this difference is greater than the threshold (defined a priori), then it is considered an active region, otherwise it is classified as background.

However, only the direct application of one of these algorithms was not enough, because the output of these algorithms contains a lot of noise. As such, to attenuate such noise a threshold has been used to eliminate spurious regions. In other words, the elimination of regions with an area below a given threshold was preceded, as was the elimination of regions on and above the river flow line.

2) Fish pairing: After the image processing for fish detection, it was necessary to perform the pairing so that it is possible to determine the correspondence between fish in consecutive images, in order to follow up. For this it was necessary to resort to temporal information, which aims to discard outliers (active regions that do not correspond to fish) through the intersection area. This information makes it possible to assess whether a region should be considered outlier.

- **Spatial information** for the pairing of fish it is very important to understand when it should be done, because spatial information is based on the hypothesis that fish travel small distances between consecutive frames and, therefore, there should be a large overlap of their corresponding active regions. This uncertainty in fish pairing occurs because in noisy environments, many active regions are detected, most of which are associated with noise. As such, we used the Mutual Favorite Pairing (MFP) algorithm as follows:

  This algorithm consists of calculating the matching, or overlap between regions detected in consecutive images. The matching operation follows a forward-backward procedure. That is to say, for a region R_t (a region detected at the instant of time t), the overlap is calculated with the region R_{t+1}. If the overlap is greater than a given threshold T, the forward step is accomplished by following the backward step (otherwise the association is not achieved). The backward step follows the same strategy as the forward step, but reverses the roles of R_t and R_{t+1}.

The pairing regions begins by automatically excluding all outliers that are furthest from the region considered as the active region. For the regions that intersect, proceed as follows. A region is only paired with your mutual favorite if the following conditions are true:

$$O_t \geqslant \alpha R_t$$

$$\wedge$$

$$O_t \geqslant \alpha R_{t-1}$$

Where O_t is given by:

$$O_t = R_{t-1} \cap R_t \quad (6)$$

If the overlap (O_t) is equal to or greater than a certain percentage (α) of the current region (R_t) and the previous region (R_{t-1}) pairing occurs, if one of the cases does not check then there is no pairing and the process continues until you find "the best" region.

3) Fish trajectories: The trajectories of the fish are very important because it is possible to collect information on the behaviour of the fish in function of the variations of the flow.

After detection of the active regions (the fish), a bounding box is generated, that is, a minimum perimeter bounding box. However, there are cases where these bounding boxes aggregate more than one fish, in the case where they swim in shoal.

After insertion of these bounding boxes, the centroids (midpoint of the bounding box) of each of them, individually, were stored so that the trajectory of each fish was differentiated. Subsequently, along the sequence of images, new centroids are being added in cases where there is a pairing of bounding boxes. If no pairing occurs, new trajectories are created.

C. Determination of the water line

One of the important tasks in this work is the determination of the water line of the river flow. The experimental channel that was filmed is an aquarium that has little ripple. As such, it was detected as a horizontal straight line.

Two different methods were applied to determine the water line in order to identify which one provides the best results. In the algorithms that are going to be studied, both have an edge detection phase, which is done by the use of the Canny algorithm:

The Canny Edge Detector, developed by John F. Canny [10] in 1986, proposes a rigorous method of detecting edges.

This process can be divided into four different steps. Initially the Gaussian filter is applied in order to remove the noise. Then the image intensity gradients are calculated before removing the maximum locations to eliminate the false response to

edge detection. Finally, a hysteresis limit is applied to remove any edges that are weak and not bound to strong edges.

After application of Canny edges detector the following solutions are considered:

First solution: To proceed to the detection of the straight line, corresponding to the flow limit, we used the Hough Transform, receiving as input the result of the application of the above-mentioned edge detector:

The Hough transform, proposed by Paul Hough [11], allows the detection of geometric shapes in a digital image. In this transformation, each point (x, y) is parameterized as follows:

$$\rho = x\cos(\theta) + y\sin(\theta) \tag{7}$$

where ρ represents the distance from the line to the origin, θ is the angle between the x-axis and the line between the origin and the nearest point of origin.

After execution of the Hough transform it is necessary to determine the local maximums of the accumulators, where each maximum corresponds to a line detected in the original image. Next, we intend to determine which horizontal line is in agreement with the greater number of lines detected. Since the flow of the river was filmed under favourable conditions, that is, in a calm environment without much agitation, a solution has been devised that passes through the identification of an approximately horizontal line. In this way you can do a search for the horizontal line that contains more number of inliers. This solution had as starting basis the existence of a large number of inliers along a horizontal zone.

The method used was based on the generation of a band, of small height, in order to count the number of inliers. This band was initialised in the lower zone of the image (excluding the zone of the base of the aquarium) and increasing its position in the vertical coordinate of the image. This process is terminated as soon as the algorithm finds a more consensual straight line model (i.e., with a greater number of inliers). After this, the representative line of the flow line is generated by joining the midpoint of the leftmost to the rightmost midpoint.

Second solution: The second solution used to accomplish the above goal is to use dynamic programming [12]. Such as the first solution, the Canny detector was first used to detect edges. After this the Gaussian lowpass filter was applied, which allows the attenuation of high spatial frequencies of the image, that is, it smoothes the image that results from the Canny detector. After this processing, the algorithm of the dynamic programming was applied to estimate the curve that crosses the image from one side boundary to the other through the largest number of edges. The dynamic programming algorithm is described below.
Dynamic programming is an iterative algorithm that allows solving a complex problem by dividing into simpler subproblems. These subproblems are solved based on the previous subproblems, saving computing time, and the intermediate solutions are stored in a cost matrix. Each of the solutions of the subproblems is indexed in order to facilitate the reconstruction of the global solution, starting from the solution of the final subproblem with less cost.
Initially, an initial value is chosen for $x_1 \epsilon D_r$ (x_1 represents the initial value of the path to be minimized, and D_r represents the set of values that x_t can take) and the optimal path starts at that position. This is achieved by changing the first column to 0, that is, $\varepsilon_i(x_i) = 0$. After this procedure two steps, forward step and backward step are applied. The first one calculates the optimal costs starting at y_1 and ending in y_N, using (8):

$$\phi(x_i, y_j) = arg \min_{\rho \epsilon D_r} d(\rho, x_i) + \varepsilon_{j-1}(\rho) \tag{8}$$

$d(\rho, x_i)$ represents the distance between x_i and the possible position ρ and ε represents the optimal cost associated with this transition and is calculated as follows:

$$\varepsilon_j(x_i) = e_{MAP}(x_i, y_j) + \min_{\rho \epsilon D_r} \left[d(\rho, x_i) + \varepsilon_{j-1}(\rho) \right] \tag{9}$$

Where e_{MAP} represents an array $M \times N$

After this first step, the average of the points in y of the line generated, from the first frame, is saved and updated from frame to frame.

IV. EVALUATION

A. River flow

The video sequence used consists of 18136 frames, but to improve efficiency, this sequence was divided into three distinct sequences. This division was made considering the sequences most relevant to the solution, that is, in the sequences where there is greater movement by the fish and greater instability of the water. Playing the entire video sequence to analyse the average height of the water line (see figure 1) it is possible to verify that between the frames 2000 - 8000 approximately it is where the greatest variation of the flow exists, or it is in the second sequence of images that the greatest instability of water is found.

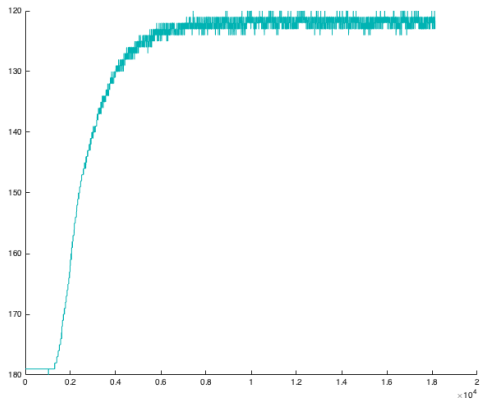


Fig. 1. Graph representing the variation of the flow rate over the entire video sequence

Two different algorithms were applied to determine the flow of the river in order to evaluate which of them can more accurately represent the flow limit.

To evaluate the performance of the algorithms, it was necessary to extract the coordinates of the river flow limit manually. This manual operation is intended to collect the ground truth, so that the algorithms are compared in this reference base.

The extraction was performed every 50 frames in the three different frames sequences. For each of these sequences the mean square error was determined:

$$MSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}$$

Where y represents the values set manually and $\hat{y}$ represents the value estimated by the algorithm.

The graphs of the figures 2 and 3 show the results obtained in the determination of the mean square error.

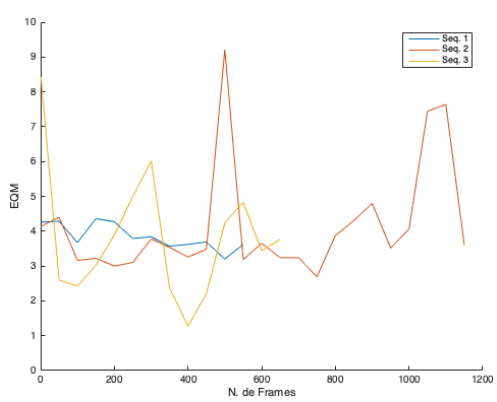


Fig. 2. Mean square error graphs on the dynamic strip

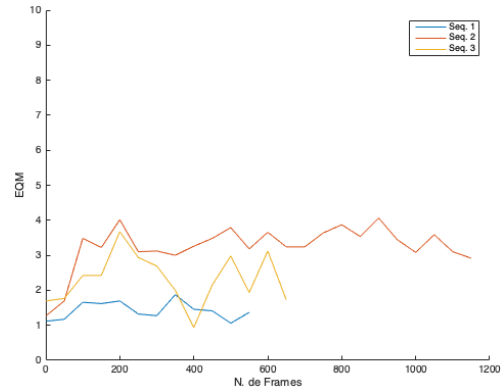


Fig. 3. Mean square error graphs on the dynamic programming

TABLE II

AVERAGE MEAN SQUARE ERROR

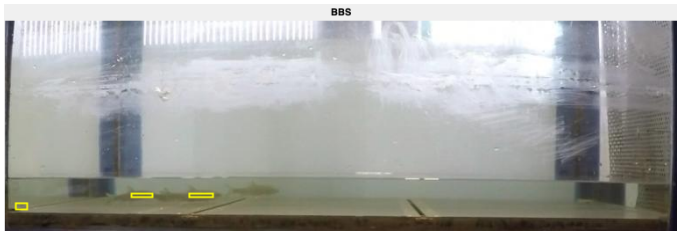
	First sequence	**Second sequence**	**Third sequence**
Dynamic strip	3.8492	4.1475	3.8199
Dynamic programming	1.4199	3.2505	2.3193

By analysing the graphs of the figure 2 and 3 and the table II it is possible to verify that the algorithm that obtained the best results was that of the dynamic programming. In the algorithm of the dynamic strip is where there is greater irregularity in the values, especially in the second sequence of frames. This is due to the fact that this sequence has, as already explained above, the greatest variation of the river flow limit, and can thus conclude that this algorithm works poorly for cases where there is low regularity. On the other hand, dynamic programming, although also having values increased in the second sequence, these can still be more stable and lower than the dynamic strip algorithm.

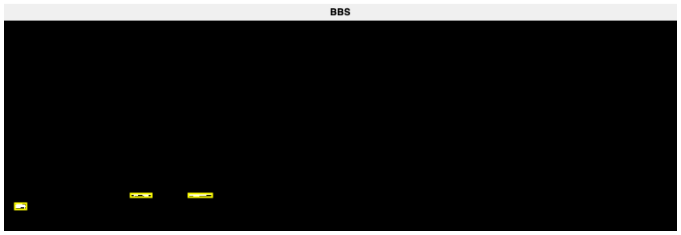
B. Fish Tracking

1) Fish detection in the image: We first determined the active regions of the fish, using the simpler proposed algorithm, Basic Background Subtraction (BBS). This was implemented and tested, but the results obtained were not sufficient to follow the fish, as can be seen in the figure 4. Therefore, the second proposed algorithm, Single Gaussian Model (SGM), was implemented. This algorithm, compared to the BBS, was able to obtain results that were much more satisfactory than the previous one, since the active regions in the second algorithm represented the fish better, as it is possible to visualise in the figure 5. It should be noted that to obtain a statistical study that allowed a comparison between the two algorithms it was necessary the manual delineation of the bounding boxes of the fish. Although this task is very time-consuming, such a statistical study would have to account for situations of (1) split (2) merge (3) merge-split. This study is somewhat complex and therefore does not fit into the context of this work, since it requires the study of

multiple interpretations in the disposition of the regions. For more details we suggest reading the article [13].



(a) Original image



(b) Binary image

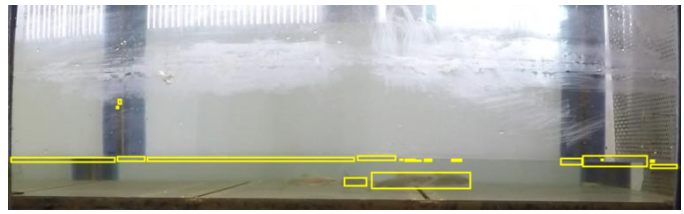
Fig. 4. Results of the application of the BBS being represented in yellow the active regions.

(a) Original image

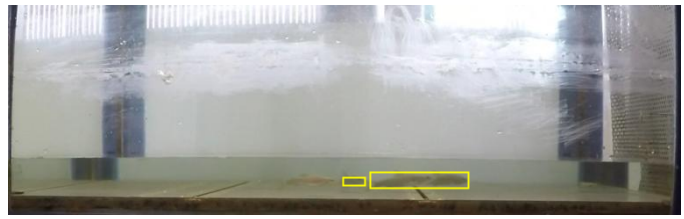
(b) Binary image

Fig. 5. Results of the SGM application being represented in yellow the active regions.

However, although the BBS can detect the active regions, it also identified many outliers (see figure 6 (a)). For this, it was necessary to reduce them by eliminating regions with very small areas, as well as regions that were on and above the river flow line. After this treatment, most of the outliers were eliminated (see figure 6 (b)) thus achieving a more rigorous fish follow-up.



(a) Original image



(b) Binary image

Fig. 6. Representative images of the before and after the removal of the outliers, the active regions being represented in yellow.

2) Pairing of active regions: After recognising the fish it was necessary to do their pairing.

Initially, for the first frame the positions of the detected active regions were stored in a memory-based, cell array data structure. For the following frames it was necessary to evaluate if the area of intersection of the previous region with the current one had a considerable value. If this intersection area was considerable, the position of the current region was added to the data structure of the corresponding region. Otherwise a new trajectory was added to the data structure. In the end the result consisted of a high number of trajectories, sometimes with very small trajectories. In the figure 7 we can observe the trajectory, represented in green:

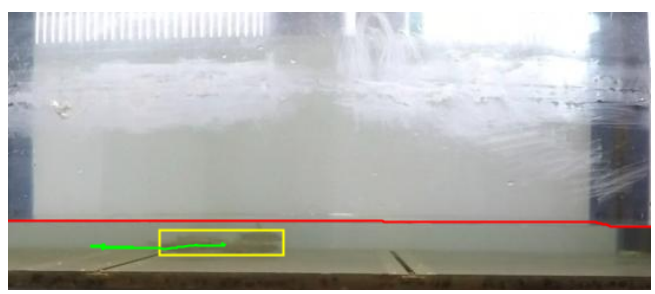


Fig. 7. Image representative of the flow line in red, from the bounding box of the fish to yellow and the trajectory of the fish to green.

In order to be able to evaluate the velocity of the fish with the increment of the flow, the following expression was used for the velocity:

$$v = \sqrt{(x_{t-1} - x_t)^2 + (y_{t-1} - y_t)^2} \tag{10}$$

Then, the fish velocity was compared to the water line, and the following results were obtained:

The first and third sequence of images are the ones with the lowest variation of river flow, while the second is where there is greater variation. In the graphs 8 and 9 the results of the velocity of the fish in the three sequences of images

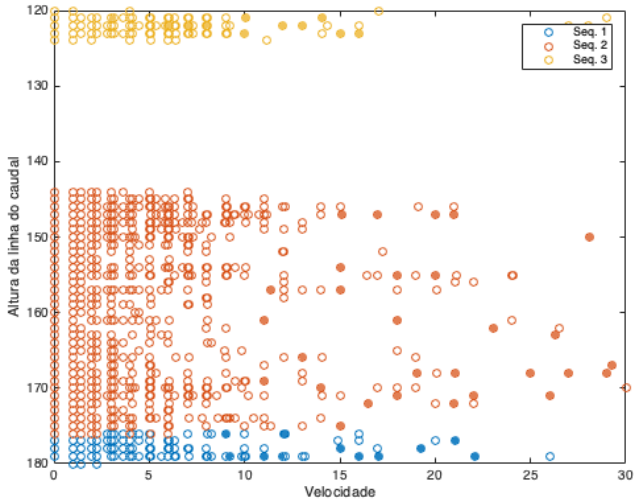


Fig. 8. Graph representing the fish velocity, as a function of the flow, along the three distinct sequences

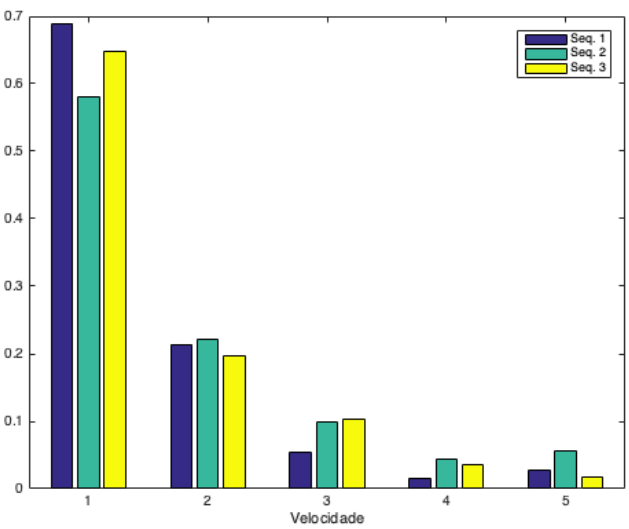


Fig. 9. Histogram representative of fish velocity along the three distinct sequences

are presented. By analysing the graph 8 and histogram 9 it is possible to verify that the highest velocities are recorded in the second sequence of images, which in turn is where variation of the flow occurs. These results suggest that the mean velocity of the fish will have increased due to the variation of water and consequent disturbance in the profile line/flow boundary.

The graph 8 has also represented the sprints and drag of the fish, the latter being represented by the filled points. By analysing these points it is possible to verify that these are the ones that present greater speed, in relation to the sprints (points not fulfilled).

V. CONCLUSIONS

Image processing continues to be complex and demanding today due to a multiplicity of factors such as detection in noisy environments, abrupt movement of fish and colour/-texture variations of moving objects. These factors have a direct impact on the solution of this project since, although the video was filmed under favourable conditions, the problematic factor remains, and with this, the difficulty in the rigorous detection of the fish as well as their accompaniment is difficult. The detection of outliers has also become one of the difficulties in the pairing of the active regions and therefore in the follow-up of the fish.

In this work the fish behaviour along the river flow rate was studied and, as a result of this work, it was verified that the fish, on average, have higher velocities when there is variation of the flow. It was also verified that the drags caused by the variation of water in the experimental flow have on average speeds higher than the sprints. This leads to wanting that the water variation in the flow affects the voluntary behaviour of the fish, causing them to suffer drag.

A. Future Work

In terms of future work, there are many lines of development that can be followed. The objective of this research work was to follow fish, in other words, to monitor their trajectories. However, there are other fish behaviour metrics that allow to distinguish statistically the effects of the different flow cycles. These behaviour metrics separate into counts and time duration. In terms of counts are considered the attempts to enter shelters and jumps, already in relation to the time have taken into account the permanence in the shelters as well as the stay next to the walls of the flow. Through these metrics individual and collective behaviours can be distinguished in relation to the experimental channel and shelters.

ACKNOWLEDGMENT

The author would like to thank the supervisor of this work, Professor Jacinto Nascimento and Professor Carlos Santiago for the guidance and availability. A special thanks to Diogo Torres, who supported me during this work. And thank you to all the people that somewhat contributed to this work.

REFERENCES

[1] J. Delcourt, M. Denoël, M. Ylieff, and P. Poncin, "Video multitracking of fish behaviour: a synthesis and future perspectives," *Fish and Fisheries*, vol. 14, no. 2, pp. 186–204, 2013.

[2] S. B. Rosenthal, C. R. Twomey, A. T. Hartnett, H. S. Wu, and I. D. Couzin, "Revealing the hidden networks of interaction in mobile animal groups allows prediction of complex behavioral contagion," *Proceedings of the National Academy of Sciences*, vol. 112, no. 15, pp. 4690–4695, 2015.

[3] C. Beyan and R. B. Fisher, "A filtering mechanism for normal fish trajectories," in *Pattern Recognition (ICPR), 2012 21st International Conference On*. IEEE, 2012, pp. 2286–2289.

[4] ——, "Detecting abnormal fish trajectories using clustered and labeled data," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE, 2013, pp. 1476–1480.

[5] X. Li, Z. Wei, L. Huang, J. Nie, W. Zhang, and L. Wang, "Real-time underwater fish tracking based on adaptive multi-appearance model," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018, pp. 2710–2714.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[7] C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. B. Fisher, "Detecting, tracking and counting fish in low quality unconstrained underwater videos." *VISAPP (2)*, vol. 2008, no. 514-519, p. 1, 2008.

[8] K. Fukunaga, "Introduction to statistical pattern recognition 2nd edition," *Academic P ress, New York*, vol. 1, no. 990, p. 43, 1990.

[9] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[10] J. Canny, "A computational approach to edge detection," in *Readings in Computer Vision*. Elsevier, 1987, pp. 184–203.

[11] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," in *Readings in computer vision*. Elsevier, 1987, pp. 714–725.

[12] C. Santiago, J. C. Nascimento, and J. S. Marques, "Fast segmentation of the left ventricle in cardiac mri using dynamic programming," *Computer methods and programs in biomedicine*, vol. 154, pp. 9–23, 2018.

[13] J. C. Nascimento and J. S. Marques, "Performance evaluation of object detection algorithms for video surveillance," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 761–774, 2006.