Automatic Fish Counting in Aquariums

Francisco João Lourenço Silvério

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. João Nuno de Oliveira e Silva
Dr. José Almeida Cruz

Examination Committee

Chairperson: Prof. Nuno Cavaco Gomes Horta
Supervisor(s): Prof. João Nuno de Oliveira e Silva
Member of the Committee: Prof. Paulo Luis Serras Lobato Correia

May 2016

To the ones I love

"This is not the end, Not even the
beginning of the end, But the end
of the beginning."

- Sir Winston Churchill

Acknowledgments

I would first like to express my deep gratitude to Champalimaud Foundation Centre for the Unknown for having given me the opportunity of developing such an interesting and motivating project.

Also, I would like to take this chance to demonstrate my most sincere gratitude and deep consideration to my extremely dedicated supervisors. Professor João Nuno Silva who was essential in accepting me as a Master student of his, for enlightening me with his precious insights, guidance and always act in my best interest. Dr. José Cruz, who was crucial in guaranteeing the cooperation with Champalimaud Foundation, for his extremely important technical inputs in this project as well as his availability to discuss any difficulties that eventually came up.

Ana Catarina Certal, Joana Monteiro and all the staff at the Champalimaud Foundation Fish Facility for their constant availability and cooperation in video recording of the fish.

Ricardo Ribeiro, Carlos Mão-de-Ferro and Alexandre Laborde, who I now consider as friends, for being so supportive and cooperative during my collaboration at the Neurosciences' Software Platform.

Another special word of thankfulness directed to the friends I had the possibility to make at Instituto Superior Técnico. I do thoroughly believe that all those memorable moments and adventures we shared will never perish in time. Among these friends, there are unique people who deserve a special mention for the deep friendship. Miguel Cirne with who I had the privilege to meet at the very beginning of my presence at Instituto Superior Técnico and work in extremely demanding projects in which we could always succeed with deep commitment and dedication. I did also have the opportunity to know and work with Miguel Fernandes with who I fought so many difficult battles throughout our engineering degree. The three of us could learn and grow together as engineers but, above all, as persons and I believe that this friendship will last.

I take this opportunity to express my most deepest gratitude to my special one, Ana Ferreira. Words are not enough to describe how special and important she was during my degree, giving me her support, patience, unconditional love and by demonstrating her extreme belief in me when I needed it the most.

Finally, I conclude with a few words for my family. First I would like to express my gratitude to my brother João for representing an example for me of the engineer I would like to be one day, for the technical help he could provide me during the past years and for being there for me as an older brother. To my sister Joana, who always stood beside me as a second mother from the very beginning of my academic path and for the constant encouragement and motivation she could always transmit to me. Por último escrevo umas palavras para a minha mãe, Irene assim como para o meu pai, Fernando. É aos meus pais que devo tudo aquilo que consegui alcançar até hoje tanto a nível pessoal como profissional. Sem o seu apoio, incentivo, exemplo, sacrifício, plena confiança nas minhas capacidades e a sua preocupação constante com a minha educação, não teria sido possível concluir esta tão hérculea etapa. É desta forma, agradecendo e com orgulho, que lhes dedico esta dissertação. Obrigado.

Resumo

Fish facilities por todo o mundo recorrem a peixes zebra (*zebrafish*) para efectuar experiências. Ao longo da sua vida, os peixes zebra são mantidos em aquários e necessitam de ser contados, manualmente, o que constitui uma tarefa morosa e que está sujeita a erros.

Este trabalho apresenta um sistema automático de contagem de peixes e um algoritmo baseado em visão computacional para realizar a tarefa. O algoritmo utiliza técnicas standard de visão computacional como subtração de fundo, isolamento de *blobs* e compensação de reflexões de forma a realizar uma contagem mais precisa. Dado que as *fish facilities* têm múltiplas categorias de peixes, o algoritmo proposto foi calibrado para garantir os melhores resultados para mais do que uma categoria.

Este projecto inclui também o design, desenvolvimento e implementação de um sistema de *hardware* completo com o objectivo de adquirir videos de peixes dentro de aquários. Experiências realizadas em amostras de video reais demonstram que o algoritmo proposto estima com sucesso o número de peixes num tanque com uma margem de erro de aproximadamente 15% do número real. Relativamente ao tempo de execução do algoritmo, concluiu-se que o tempo necessário para estimar o número de peixes é, em geral, inferior ao tempo que técnicos especializados demoram a realizar uma contagem manual.

Como trabalho futuro, a identicação de diferentes tipos de fenótipos assim como a contagem do número de macho e fêmeas em aquários, são exemplos de ideias interessantes que podem ser implementadas neste tipo de aplicações.

Palavras-chave: visão computacional, contagem de peixes zebra, subtração de fundo, optical flow, momentos de Hu, processamento de imagem

Abstract

Fish facilities around the world resort to zebrafish to perform experiments. During their lifetime, fish are maintained in tanks and need to be manually counted representing a time consuming and error prone task.

This work presents an automatic fish counting setup and a computer vision based algorithm to perform the task. The algorithm uses image processing methods such as background subtraction, blob isolation and mirroring compensation to perform a more precise fish counting.

Since fish facilities have multiple categories of fish, the proposed algorithm was calibrated to guarantee the best results for more than one category. Moreover, this project does also comprise the design, development and implementation of a full hardware setup for the video recording of fish inside tanks.

Experiments on real video samples show that the proposed algorithm can successfully estimate the number of fish in a tank with an error margin below 15% of the real fish number inside a tank.

Regarding algorithm execution time, we could conclude that the time needed to output an estimate of the number of fish in a tank is lower than the manual counting performed by technicians.

In future work, the identification of multiple phenotypes as well as counting the number of male and female inside tanks, are examples of very interesting ideas which should be implemented in this type of solution.

Keywords: computer vision, zebrafish counting, background subtraction, optical flow, Hu moments, image processing

Contents

List of Tables

List of Figures

Chapter 1

Introduction and scope

1.1 Overview

Zebrafish (*danio rerio*) is a small freshwater fish that is widely used as an animal model in biomedical research with origins in specific locations across the globe as we can see in Figure 1.1 [1]. Research laboratories around the world require a huge number of individuals to perform a great variety of experiments. Those fish are breed and maintained in big fish facilities managing hundreds to thousands of fish tanks. Usually these tanks are standardized containers (for instance, from 3 to 8 l) which may host several dozens of animals each.

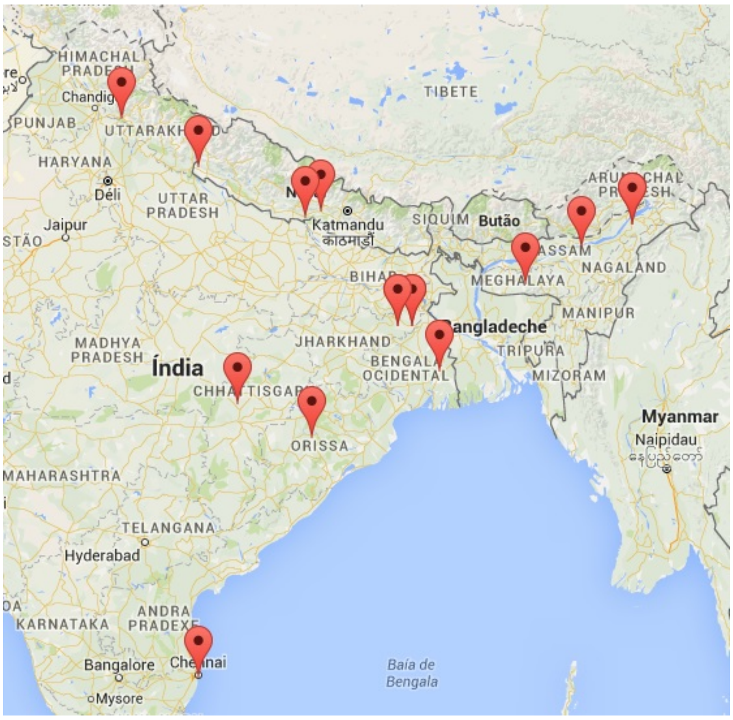


Figure 1.1: Geographic Distribution of *danio rerio*. (www.fishbase.org)

Obtaining an up to date count of the total number of animals in a fish facility is an essential task, performed by human technicians who manually extract animals with the help of small fish nets. This manual counting process requires a significant amount of time and is error prone. Moreover, handling animals for counting induces significant stress, with all the harmful consequences that may cause to the animals and, consequently, affect the scientific experiments they are involved in.

Finding a noninvasive automatic procedure to obtain the precise number of zebrafish in facilities tanks, avoiding all the disadvantages of manual counting, is a long sought goal of fish facilities managers.

Fish counting automation may be approached using computer vision. In fact, today there are many examples of complex applications making use of computer vision techniques [2] such as:

- Optical character recognition: reading handwriting and automatic number plate recognition
- Machine inspection: measure tolerances on aircraft wings or inspect steel casting with X-ray vision
- 3D model building: 3D models from aerial photographs
- Medical Imaging: perform long-term studies of brain morphology
- Face detection: to be used in image searching
- Visual authentication: grant people permission for accessing buildings based on morphology features
- People tracking: monitor passenger motion in airports.

Some of the previous applications make use of techniques such as optical flow and background subtraction which have particular interest in this project. Optical flow [3] is highly related to pixel motion and its variability between frames. Furthermore, it gives a reliable estimation of displacement between different frames. Background subtraction is specially relevant when, for example, the need for isolating moving regions in a sequence of images arises. Those techniques will be explained in more detail in Chapter 3. In this fish counting project, since images are two-dimensional, the main difficulties to overcome are: regions where fish overlap, mirroring effect, i.e., fish reflections on the tanks sides and, mainly, fish shoaling.

1.2 Environment

Fish facilities usually use zebrafish of the following genotypes: wild type zebrafish from strains AB and TU, mutant Nacre zebrafish and transgenic fish from multiple lines with TU and Nacre background. Genotype may be defined as the genetic makeup of an organism that determines specific physical characteristics, phenotype, of an individual. It is important to state that to the different genotypes correspond different phenotypes.

From this point, we will assume that fish are divided into four different categories with different genotype and age: AB/TU fries (30-day old), AB/TU adults (90-day old), Nacre fries (30-day old) and Nacre adults (90-day old). The physical differences between the categories Nacre and AB/TU can be seen from Figure 1.2 to Figure 1.5. This type of fish is extremely sensitive when changes in its environment occur, thus, inflicting too much stress while handling may, in some cases, lead to death.

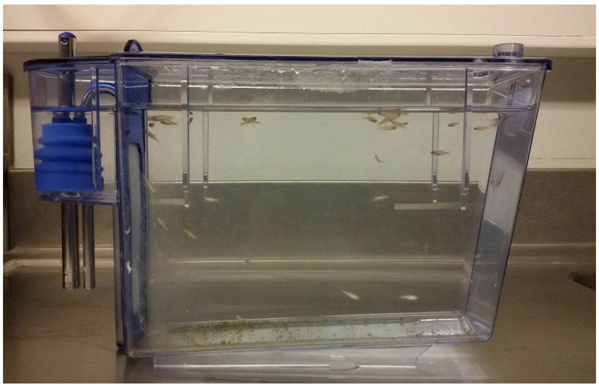


Figure 1.2: Nacre zebrafish fries.

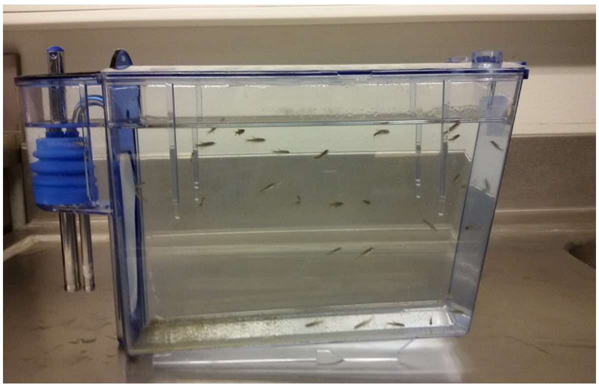


Figure 1.3: AB/TU zebrafish fries.

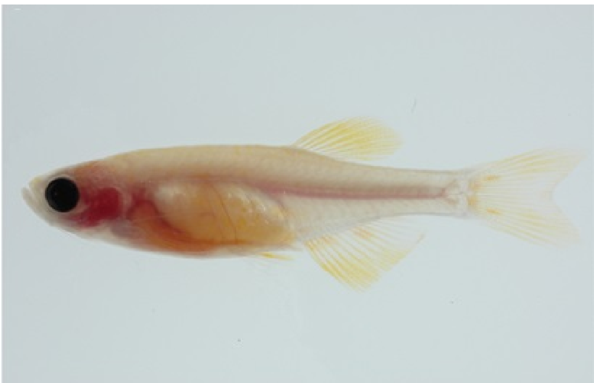


Figure 1.4: Adult Nacre zebrafish. [4]

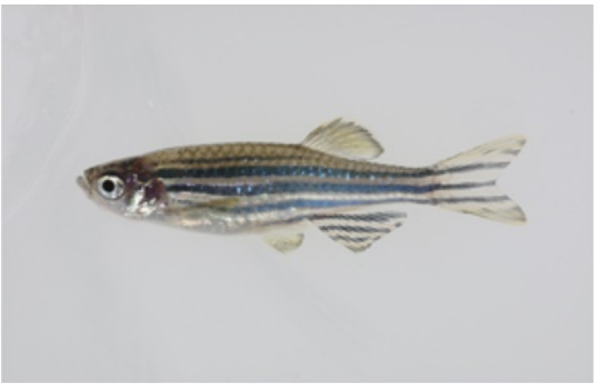


Figure 1.5: Adult AB zebrafish. [4]

Fish growth rate depends on factors such as fish density in a tank and feeding. In this project, fish have an average length (fork lenght) of 1.8±0.16 cm (fries) and 2.5±0.15 cm (90-day old adults) and, as adults, should not be longer than 3 cm.

At the fish facility, fish are maintained in standard sized tanks with, for instance, 3.5L, as it can be seen in Figure 1.6, where water temperature is around 28°C (27.94±0.05) and the external temperature is approximately 25°C.

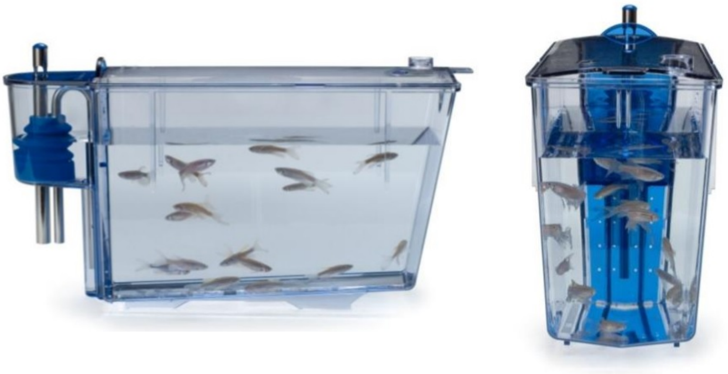


Figure 1.6: Zebrafish tank type. [5]

Tanks are stored, side by side, in appropriate housing systems, where the water recycling system makes sure that the water in aquariums is constantly being renewed and the automatic feeding system provides the correct periodic feeding for each tank. Figure 1.7 shows an example of a housing system where that is verified.

In our project, the number of fish per tank may vary from less than a dozen to a maximum number of 35. At ”day zero” fish eggs are placed inside tanks after microscopic larvae counting. Figures 1.8 and 1.9 show the physical features of a zebrafish larva at day zero for the two genotypes. After 30 days, a manual counting is performed to analyse how many fish survived the first month allowing the extraction of data to calculate the mortality ratio. This procedure is repeated on the third month when zebrafish are adults.

To have an idea about the time that the manual counting process usually takes, we can analyse Table 1.1. By analysing the table, we can conclude that there is no relationship between the number of fish and the time spent to perform the task. In those counts, the lowest value obtained was 28 seconds for 12 fish in a tank. These values can be used as a reference to compare to the time performance of our algorithm that will be done in Chapter 5.

Fish Number	12	16	18	19	23	23	26	26	27	27	28	28	28	28	29	31
Time [s]	28	40	35	48	58	44	55	78	67	91	53	56	69	52	43	82

Table 1.1: Counting times obtained in 16 different counts for zebrafish fries performed by an accredited technician.

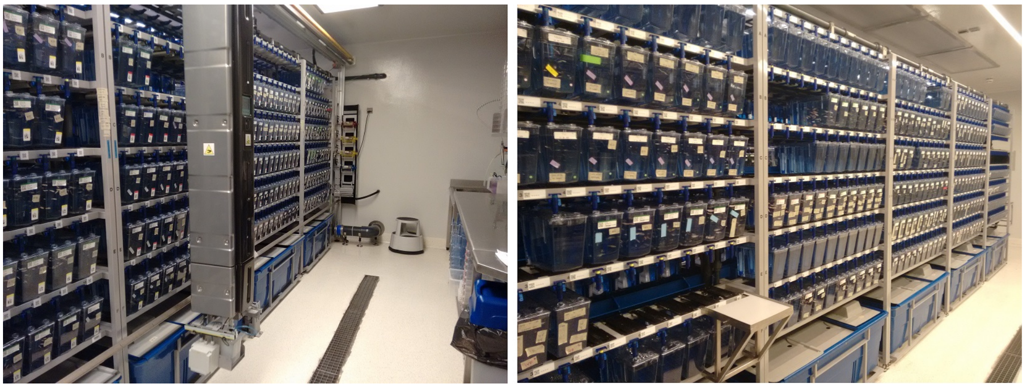


Figure 1.7: Housing system at the fish facility

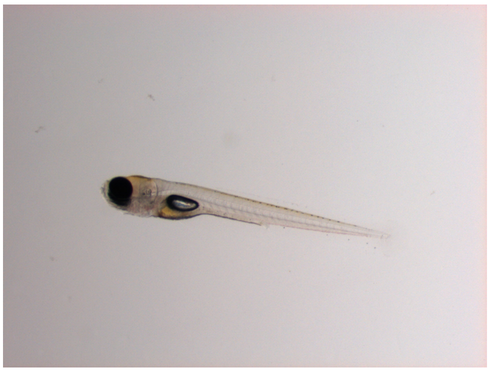


Figure 1.8: Nacre zebrafish larva.

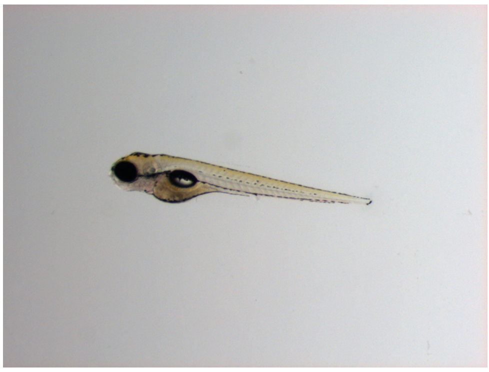


Figure 1.9: AB zebrafish larva.

1.3 Results

In this project, an experimental setup for video recording and the development of an algorithm for zebrafish counting in tanks that combines background subtraction, blob counting and mirror compensation are proposed.

Hence, the main results of this project were:

1. Building a full prototype (hardware and software) of a counting device.
2. Counting the number of fish in a tank with a maximum error margin of 15% of the real fish value inside it given by manual counting.
3. Ensuring that all the procedures regarding the recording of the videos do not cause stress to the animals (noninvasive technique).
4. Publishing a paper regarding this work in the International Conference on Image Analysis and Recognition 2016 (ICIAR 2016).

1.4 Report Structure

This report is organized according to the following division of chapters:

- **Chapter 1: Introduction and scope** - Provides an overview of the project context, defines the goals and explains how the report is organized. Describes the parameters regarding environment features.
- **Chapter 2: State of the art** - An assessment of previous research and methodologies employed, by different authors, in the development of other projects relevant in this dissertation context.
- **Chapter 3: Theoretical Background** - Comprises a general theoretical overview and provides theoretical details regarding the different computer vision and mathematical tools used: background subtraction, optical flow and Hu moments, respectively.
- **Chapter 4: Implementation** - Describes the features regarding the video recording setup and container in terms of structural specifications and illumination. Likewise, detailed information about the computational hardware specifications and the different steps of the algorithm are given.
- **Chapter 5: Evaluation** - Represents the discussion of the results obtained using the fish counting algorithm.
- **Chapter 6: Conclusions and Future Work** - Finally, the project's achievements are summarized and discussed. Recommendations on future work are also made.

Chapter 2

State of the art

Nowadays there are many different computer vision techniques used in object counting. Since the purpose of this work is to develop a fish counting system, all of the following approaches represent different solutions related to this specific problem. Typically, these systems follow common patterns such as:

- Capturing images (normally video acquisition)
- Extracting features (such as colors)
- Detecting blobs
- Provide a counting based on the blobs detected.

For instance, Y.H. Toh et al. [6] present a method of counting feeder fish through image processing techniques. A video of a school of fish is acquired and analysed to output a counting number. At some point filters are applied to background and noise of the image with already identified blobs. The number of fish contained in one image is estimated taking into consideration the individual area of the blobs in one frame and the average number of fish over all frames is calculated. It is relevant to take into account that the setup provided made possible to have a background very easily distinguished from the fish. The tank in the experiment has, apparently, low-level water (which reduces the chance of having overlapping fish), there are no referred restrictions regarding luminosity intensity and it is assumed that the fish are most likely to appear isolated. Considering all the specifications it is possible to count correctly schools of 5, 10, 15 and 50 fish. This approach is not suitable for our project since the tanks that were used in our videos had significant differences from the ones used in Toh et al. experiments such as water depth and volume.

Khanfar et al. [7] suggest the development of algorithms which allow the recognition of fish in the images and track the locations of individual fish from frame to frame. This work takes as an assumption that the majority of the fish tend to move against an approximately stationary background and so, that motion, can be useful in detection. The background is adjusted according to images with no fish. This is used to calculate a histogram for the pixel amplitude used to define thresholds to be set and then isolate regions containing fish. Once it is possible to obtain those regions, an algorithm of edge detection and region growing are used allowing an accurate counting of the number of fish. Several frames provide information for tracking: given an identified fish in a well defined location, it is expectable that, in case of a merging at the next frame in a very close location (for instance another well identified fish), the total area of the merging equals the sum of the area of the fish in it. It is also considered that the motion pattern of fish at a frame may be used to determine if regions are likely to merge in the subsequent frame. The tracking algorithm requires information such as the location of fish, intensity of the region, length, width and area of the fish. With this information it is possible to calculate an Euclidean distance used to associate one region in a frame to a region in the next frame. The procedure enabled the tracking of regions from one frame to the following and then split any new merged regions considering former positions coordinates and that a region travels with constant velocity. This project was relevant in the development of ours since, after analysing this work, we adopted the idea that when blobs merge, the total area of the merging equals the sum of the area of fish in it.

In Fabic et al. [8] work, canny edge detection algorithm is used combined with a coral-blackening background process. Video is recorded in a coral reef environment, which represent a much more complex background than, for instance, the one used by Y.H. Toh et al. The main difference introduced by the project, comparing to the aforementioned references, is that after the detection of each blob, the Zernike moment [9] of every individual blob is calculated having in consideration a standard predefined fish template (depending on the type of fish used in the experiment). Zernike moments allow the mapping of an image using complex Zernike polynomials. The orthogonality between the polynomials makes possible to represent the properties of an image with no redundancy or information overlapping between distinct moments. Despite being dependent on the scaling and translation of the object, given a region of interest, Zernike moments have the advantages of having magnitudes independent of the rotation angle of an object. Hence, they are used to describe shape characteristics of the objects. Thus, in order to identify different fish species and count them in every frame, a set of orthogonal Zernike moments is chosen and applied due to their rotational, translational and scale invariant properties. This work had particular interest in ours since it considers the creation of fish templates with features for each type of fish. This was used as a motivation in our work since we decided to do something similar with fish areas in the different fish categories.

Significant differences regarding the setup environment control are presented by Spampinato et al. [10] where the videos are recorded in open sea and reflect the changes, for instance, in the luminosity and water flowing (background variation). Inherent to the image processing tasks are sub-processing systems consisting of texture and colour analysis, fish detection and fish tracking. The analysis of the statistical moments of the grey-level histogram is the chosen approach to describe mathematically the image texture (e.g. brightness and smoothness). In colour analysis hue, saturation and pixels values are compared to predefined threshold to decide which color one region has in a frame. A moving average algorithm consisting on frame subtractions is used to provide fish detection analysing moving pixels and is made between the background image and the current frame. It is claimed that this particular algorithm has the advantage of giving a good balance between results accuracy and total processing time. On the other hand, in scenes with no static background false positives arose and had to be removed using adaptive gaussian mixture model [11] which modelled each background pixel and classified them based on the likelihood between the real value of the pixel and the one assigned by the Gaussian Mixture Model. Combining the two algorithms, it becomes possible to generate an approximation of the number of fish in a frame by applying a connected component labelling algorithm. Finally the tracking system uses an algorithm based on the matching of blob features and on a pixel histogram matching. This approach motivated the usage of gaussian mixture models in the background subtraction algorithm in our project.

Considering the setup features used in our project, Boom et al. [12] work may have special relevance due to the applied techniques, video environment and results. For instance, regarding fish detection, Gaussian Mixture Model (as presented by Spampinato et al. [10]) which allows dealing with multimodal backgrounds and Adaptive Poisson Mixture Model variant are implemented, as mixture-based algorithms. This procedure models, for each pixel, the distribution of the intensity values typically contained in the background image. However, the computational processing costs tend to increase exponentially as more models are added. Adaptive Poisson Mixture Model is used to handle illumination variations. It is also stated that another algorithm is developed to specifically deal with sudden illumination changes where the reflectance component (static element) of each frame is separated from its illumination component (which is a parameter that varies depending on the light conditions) and the new background model is then computed as a temporal median of these two components. Each pixel has a list of its 20 most recent intensity values and if the value of a pixel in a new frame matches with a high number of values on the list, the pixel is considered background in the new frame. The ViBe algorithm is the responsible for the described verification. All of the above methods are combined and employed by a trained classifier which has no great interest for our project since our algorithm does not include machine learning techniques. Filtering is applied to remove noise and isolate blobs. A post-processing detection module is used to filter bad detections and reduce false positives through the analysis of each blob by verifying if its shape, texture, motion, structure and segmentation match to expected values from correctly identified fish. Tracking is based on covariance-based models where the template of a fish is represented as the covariance matrix of a set of feature vectors computed for each pixel of the object. Pixel's (x,y) coordinates, RGB values, hue value and the mean of a grayscale histogram are included in each vector. Covariance matrices are compared to decide model similarity using Förstner's distance. The tracking algorithm is connected to the fish detection since a new fish is only tracked if there is the indication that a new one is detected. Tracked fish are located in the scene considering that the search window is based on the fish' speed and direction in the previous frame. Thus, it is possible to calculate candidate regions and compute their covariance matrix. In the end, a new location for a fish is set to the region which, according to Förstner's distance, is most similar to the fish model.

Our project presents different environment and features (such as fish size) comparing to the ones presented in the aforementioned references. Nevertheless, as already stated, the references were used as a motivation to some decisions we made in our project such as the use of background subtraction with gaussian mixture models and the acquisition of the typical area to create a template for each fish category. Combining all these informations we decided to develop an approach specifically for our application.

Chapter 3

Theoretical Background

In this chapter we present the theoretical background considered in the implementation of our algorithm. During the development of this project, the need to clearly distinguished fish from the other elements in the image emerged. Consequently, including a background subtraction algorithm was considered.

Another relevant aspect that had to be taken into consideration was the fish mirroring on the sides of the tank and surface. In this way, we searched for a method which could allow the identification of these situations and, in the end, decided to include the calculation of Hu moments for blobs.

In this section we also discuss optical flow that, in spite of not being used in the final version of the algorithm, was also applied and can be further explored in future work.

Thus, our final version of the algorithm involves the use of background subtraction for blob isolation and Hu moments calculation for consequent fish mirroring compensation and blob counting.

3.1 Background Subtraction

Background subtraction is a technique that allows the elimination of static elements in an image which can be considered as background. Since in our project we need to provide a clear identification of fish in an image by isolating them, background subtraction is an extremely relevant method to distinguish between what can be considered as background and foreground (fish).

The contents of this section are strictly based on Zivkovic and Heijden [11][13] works and gives an explanation of the idea implemented in MOG2 (”Mixture of Gaussians 2”) method implemented on OpenCV Library. One important feature of this algorithm is that it selects the appropriate number of gaussian distribution for each pixel. Reynolds defines Gaussian Mixture Model (GMM) as a parametric probability density function which is represented as a weighted sum of Gaussian component densities [14]. Likewise, GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system. The parameters for the model are estimated from training using Expectation-Maximization algorithm or, as in this MOG2 implementation, Maximum A Posteriori (MAP) estimation from a well-trained prior model.

The process of background subtraction is usually related to the identification of what will be defined as

”intruding objects” in a certain scene which, in this case, correspond to zebrafish in an image. Normally, a common assumption is that the images in a scene or, in this project, frames in a video without intruding objects, exhibit an approximately regular behaviour. It is possible to describe this regular behaviour using mathematical tools to calculate a statistical model of a scene that can be used to detect, in each moment, which regions in an image do not fit this model.

Taking into consideration that an image is the result of a combination of pixel values (RGB for example), it makes sense that each pixel has a probability density function related to the scene model. Thus, a pixel from a new image will only be considered as background if its new value is well described by its density function. In fact, given a pixel value, at time t in any color space - $\vec{x}^{(t)}$ - deciding if a pixel is part of background (BG) or foreground (FG) can be made calculating:

$$D = \frac{p(BG|\vec{x}^{(t)})}{p(FG|\vec{x}^{(t)})} = \frac{p(\vec{x}^{(t)}|BG)p(BG)}{p(\vec{x}^{(t)}|FG)p(FG)} \tag{3.1}$$

Uniform distribution is assumed for foreground objects appearance then $p(\vec{x}^{(t)}|FG) = c_{FG}$ and $p(FG) = p(BG)$ since there is, generally, no information about the foreground objects that can be seen or even if or when they will be present in an image. Given the aforementioned information, one pixel is considered as background if:

$$D = p(\vec{x}^{(t)}|BG) > c_{thr}(= Rc_{FG}) \tag{3.2}$$

where c_{thr} represents a threshold value.

The background model, $p(\vec{x}^{(t)}|BG)$, is estimated from a training set χ, which consequently allows the definition of the estimated model as $\hat{p}(\vec{x}|\chi, BG)$ since it depends on the training set. However, features in an image may rapidly or gradually change, new objects may appear or old ones may be removed. Thus, a GMM is used so that the set can be update by discarding old samples and adding new ones. For each t in a time period T, the training set is $\chi_T = \{x^{(t)}, ..., x^{(t-T)}\}$. As the samples may contain foreground objects, the estimated density is given by $p(\vec{x}^{(t)}|\chi_T, BG + FG)$ and a GMM with M components is used:

$$p(\vec{x}^{(t)}|\chi_T, BG + FG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x}, \hat{\vec{\mu}}_m, \hat{\sigma}^2 I) \tag{3.3}$$

where $\hat{\pi}_m$ represent the estimated mixing weights, $\hat{\vec{\mu}}_1$,...,$\hat{\vec{\mu}}_M$ correspond to the estimates of the mean values and $\hat{\sigma}_1^2$,...,$\hat{\sigma}_M^2$ are the estimates of the variances that describe the Gaussian components. For new data those three estimates are recursively update and if they meet certain criteria, old samples are discarded and the training set is updated.

In the end, the general steps for this GMM background algorithm are:

- Classify the new sample $\vec{x}^{(t)}, p(\vec{x}^{(t)}|\chi_T, BG) > c_{thr}$

- Update $p(\vec{x}|\chi_T, BG + FG)$

- Update $p(\vec{x}|\chi_T, BG)$ to select the components of the GMM belonging to the background

3.2 Optical Flow

Optical flow is a computer vision technique that makes possible the tracking of objects. During the development of this project, it was first thought that we could possibly track the different fish in a tank, in successive frames, in order to compensate fish overlapping.

This technique is presented by Gunnar Farnebäck work, representing a two-frame motion estimation algorithm [3] and will be next summarized. Specifically, this method uses polynomial expansion which approximates pixel's neighbourhoods with polynomials. Quadratic polynomials, particularly, are used since they give the local signal model that can be expressed in a local coordinate system by:

$$f(x) \sim x^T A x + b^T x + c \tag{3.4}$$

where A is a symmetric matrix, b a vector and c a scalar. The previous coefficients are the result of weighted least squares fit applied to the signal values in a neighbourhood. Two components compose the weighting: certainty and applicability, where certainty is connected to the signal values in the neighbourhood and applicability represents the weight of points in the neighbourhood given a certain position.

A neighbourhood of pixels is approximated by a polynomial as a result of polynomial expansion. Thus, considering the quadratic polynomial:

$$f_1(x) = x^T A_1 x + b_1^T x + c_1 \tag{3.5}$$

we can put together a new function dependent on a displacement d,

$$\begin{aligned} f_2(x) = f_1(x-d) &= (x-d)^T A_1 (x-d) + b_1^T (x-d) + c_1 \\ &= x^T A_1 x + (b_1 - 2A_1 d)^T x + d^T A_1 d - b_1^T d + c_1 \\ &= x^T A_2 x + b_2^T x + c_2. \end{aligned} \tag{3.6}$$

If we equate the coefficients in the quadratic polynomials the result is:

$$A_2 = A_1, \tag{3.7}$$

$$b_2 = b_1 - 2A_1 d, \tag{3.8}$$

$$c_2 = d^T A_1 d - b_1^T d + c_1. \tag{3.9}$$

The important remark about this equations is that 3.8 can be solved for the displacement d every time A_1 in an invertible matrix (nonsingular):

$$2A_1 d = -(b_2 - b_1), \tag{3.10}$$

$$d = -\frac{1}{2} A_1^{-1} (b_2 - b_1). \tag{3.11}$$

Assuming that an entire signal can be represented by a single polynomial and a global transition by the relation between two signals is unrealistic. Thus, we replace the global polynomial in 3.5 with local polynomial approximations. This is done with a polynomial expansion for both images, resulting expansion coefficients $A_1(x)$, $b_1(x)$ and $c_1(x)$ for the first image, or, in this project, we may refer to the images as frames, and $A_2(x)$, $b_2(x)$ and $c_2(x)$ for the second image. Then, we settle the approximation:

$$A(x) = \frac{A_1(x) + A_2(x)}{2} \tag{3.12}$$

and

$$\Delta b(x) = -\frac{1}{2}(b_2(x) - b_1(x)) \tag{3.13}$$

to obtain:

$$A(x)d(x) = \Delta b(x) \tag{3.14}$$

where the presence of $d(x)$ indicates that the global displacement, d, in equation 3.6 have been replaced with a spatially varying displacement field. Assuming that the displacement field is slowly varying, we can (discretely) integrate information regarding a neighbourhood of each pixel. In this way, we can find $d(x)$ that satisfies 3.14 over a neighbourhood I of x:

$$\sum_{\Delta x \epsilon I} w(\Delta x)||A(x + \Delta x)d(x) - \Delta b(x + \Delta x)||^2. \tag{3.15}$$

The minimum is obtained for:

$$d(x) = (\sum wA^T A)^{-1} \sum wA^T \Delta b \tag{3.16}$$

and this solution exists and is unique unless the neighbourhood exhibits the aperture problem [15]. After some calculations, it is possible to conclude that the minimum value is given by

$$e(x) = (\sum w\Delta b^T \Delta b) - d(x)^T \sum wA^T \Delta b. \tag{3.17}$$

This means, in practice, that we compute $A^T A$, $A^T \Delta b$ and $\Delta b^T \Delta b$ pointwise and use w to average those products before solving for the displacement.

The robustness is improved if the displacement field is parametrized by a motion model. We can use the eight parameter model in 2D [16]

$$d_x(x, y) = a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy, \tag{3.18}$$

$$d_y(x, y) = a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2 \tag{3.19}$$

and write the equations as

$$d = Sp, \tag{3.20}$$

$$S = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{bmatrix}, \tag{3.21}$$

$$p = (a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8)^T. \tag{3.22}$$

If we update 3.15 with the previous equalities we get

$$\sum_i w_i ||A_i S_i p - \Delta b_i||^2, \tag{3.23}$$

where i represents the coordinates in a neighbourhood. The solution is then

$$p = (\sum_i w_i S_i^T A_i^T A_i S_i)^{-1} \sum_i w_i S_i^T A_i^T \Delta b_i \tag{3.24}$$

It can once more be verified that we can calculate $S^T A^T A S$ and $S^T A^T \Delta b$ pointwise and then average these values with w.

This method is not restricted to comparing two polynomials at the same coordinates. In case *a priori* information regarding the displacement field is known, the polynomial at x in the first signal can be compared to the polynomial at $x + \tilde{d}(x)$ in the second signal, where $\tilde{d}(x)$ represents the *a priori* displacement field. Having this into consideration, we can replace 3.12 and 3.13 by

$$A(x) = \frac{A_1(x) + A_2(\tilde{x})}{2} \tag{3.25}$$

$$\Delta b(x) = -\frac{1}{2}(b_2((\tilde{x}) - b_1(x)) + A(x)\tilde{d}(x) \tag{3.26}$$

where

$$\tilde{x} = x + \tilde{d}(x). \tag{3.27}$$

3.3 Hu Moments

During the development of this project, we were able to realize that zebrafish tend to be mirrored in some areas of the tank such as the right side of the tank and the surface. Since we perform blob counting in our algorithm, that will be further detailed in Chapter 4, counting blobs resulting from mirroring influence the final counting estimate. In this way, we used a mathematical tool that allow us to identify those situations and implement the necessary compensating measures.

Taking that into consideration, in this thesis we compute Hu moments in order to compensate some of the fish mirroring effects. This mathematical procedure was developed by Ming-Kuei Hu [17] in his work regarding moment invariants in recognitions patterns. His work presents a theory of two-dimensional moment invariants for planar geometric figures. There, absolute orthogonal invariants are derived and are used in our project to get the pattern identification of similar shape independent of size, position and orientation. The expressions were derived from algebraic invariants applied to the expression that describes a moment, consisting on groups of nonlinear centralised moment expressions. This resulted in absolute orthogonal moment invariants that can be used for orientation independent pattern identification. For the second and third moments, six orthogonal invariants are derived:

$$\begin{aligned}
&\mu_{20}+\mu_{02},\\
&(\mu_{20}-\mu_{02})^2+4\mu_{11}^2,\\
&(\mu_{30}-3\mu_{12})^2+(3\mu_{21}-\mu_{03})^2,\\
&(\mu_{30}+\mu_{12})^2+(\mu_{21}+\mu_{03})^2,\\
&(\mu_{30}-3\mu_{12})(\mu_{30}+\mu_{12})[(\mu_{30}+\mu_{12})^2-3(\mu_{21}+\mu_{03})^2]\\
&+(3\mu_{21}-\mu_{03})(\mu_{21}+\mu_{03})[3(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2],\\
&(\mu_{20}-\mu_{02})[(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2]+4\mu_{11}(\mu_{30}+\mu_{12})(\mu_{21}+\mu_{03}),
\end{aligned}\tag{3.28}$$

and one skew orthogonal invariants,

$$\begin{aligned}
&(3\mu_{21}-\mu_{03})(\mu_{30}+\mu_{12})[(\mu_{30}+\mu_{12})^2-3(\mu_{21}+\mu_{03})^2]\\
&-(\mu_{30}-3\mu_{12})(\mu_{21}+\mu_{03})[3(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2].
\end{aligned}\tag{3.29}$$

This skew invariant is useful in identifying mirror images and is specially relevant for this thesis as well as the first invariant. In fact, this method can be generalized to accomplish pattern identification not only independently of position, size and orientation but also independently of parallel projection which is used in this thesis and will be further detailed in Chapter 4.

Chapter 4

Implementation

The development of our technique involved the creation of a specific recording setup prototype for this particular application where environment features, described in Section 1.2, were taken into consideration. After obtaining video sequences with fish, it is then possible to start the video processing and estimate the number of fish inside the tank. In this chapter we detail the development of the recording setup, from the first version until the most recent one and relate the theoretical background presented in Chapter 3 to the developed counting algorithm.

4.1 Danio Recording Setup

This work included the development of a full prototype (hardware and software) of a counting device. Figure 4.1 shows the first version of the recording prototype, which we call Danio Recording Setup in honour to the scientific name of the zebrafish *danio rerio*. This version was made of a standard plastic box covered, inside, in blue musgami waterproof paper. Moreover, a strip of blue LEDs [18] was fixed to

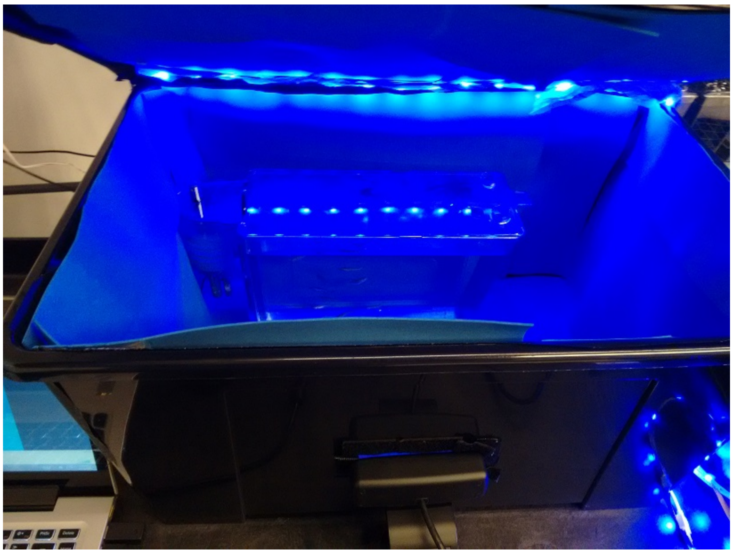


Figure 4.1: First version of the Danio Recording Setup with external USB webcam.

the cover of the box to illuminate the tank when closed.

With those specifications, it was possible to generate a blue uniform environment inside the box which was recorded by an external USB webcam, connected to a computer, from one hole in one side of the box where the camera fit in.

Given the different physical patterns exhibited by the zebrafish and the fact that the color of the tanks is bluish, it was thought that a clean blue environment could bring good image contrast and uniform quality, representing an advantage for video processing. In this way the background subtraction algorithm can rapidly stabilize and, more effectively, allow the detection of fish in the foreground.

At the very beginning of the project, an already existing setup covered inside by a textured cloth was used to make the first experiments with background subtraction. We can see an example in Figure 4.2 where a tank is placed inside a container with the textured background and in Figure 4.3 an image obtained with the first version of the setup exhibiting a clean uniform background pattern. Codification of images with textured backgrounds is harder than untextured since, in those cases, we can not exploit spatial redundancy when coding. The result are images with larger size, thus taking longer time to process, since we need more bits to code the texture in the image. By using a clean untextured background, the background subtraction algorithm can rapidly stabilize and, more effectively, allow the detection of fish in the foreground. The most remarkable disadvantage of this first setup was the variable position of the tank and webcam . Since this was a first approach for the development of the setup, the placement of the tank inside the container and the webcam at the hole in the box was not always done exactly at the same place. Hence, whenever we wanted to make new trials, we could not achieve the exact recording positions which have impact on the algorithm output since it is dependent on the distance between the camera and the tank. Thus, to ensure the repeatability of the process, we developed a new version of the prototype.

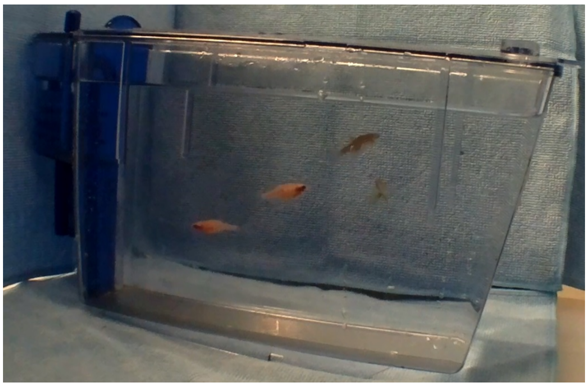


Figure 4.2: Tank inside a container with textured background.

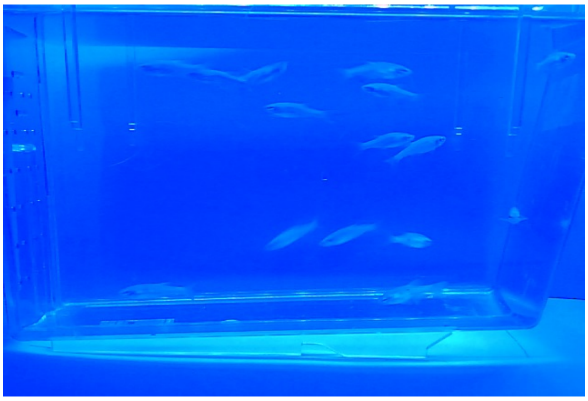
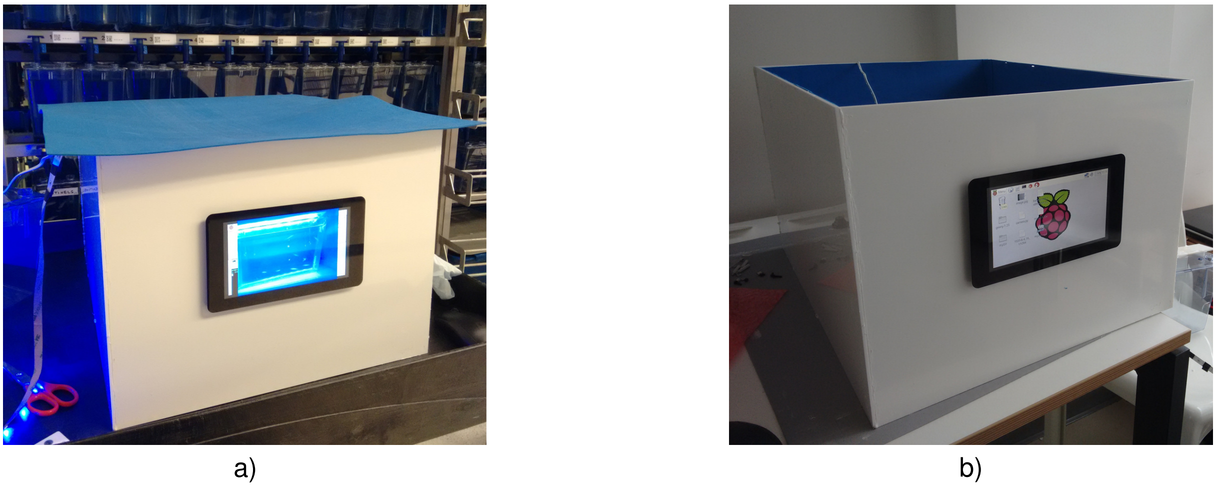


Figure 4.3: Image acquired using the first version of the Danio Recording Setup.

The most recent version of the Danio Recording Setup is shown in Figures 4.4 a) and b). This enhanced setup includes structural and hardware diferences comparing to the previous version. This container is a parallelepiped structure made of acrylic, with the dimensions presented in Figure 4.5 and is also covered in blue waterproof paper as in the first version for the same reasons. The most important differences on the inside of the container are the holes carved on the bottom which are intended to fix



a) b)

Figure 4.4: Latest version of the Danio Recording Setup (a) on at the fish facility, b) in standby mode).

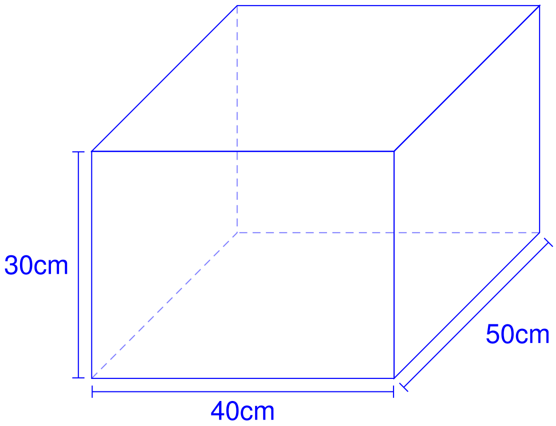


Figure 4.5: Danio Recording Setup dimensions.

each tank to it making sure that the distance to the camera is always constant and it does not influence the algorithm output.

This setup also included housing for the tanks. In Figure 4.6 we can see how each tank is fixed to the bottom. There are three different positions where a tank can be placed in making possible to record with three different distances to the fish. The same blue LEDs are fixed to a moving piece that was designed to fit on the top of each tank, at exactly the same place. Hence, the same light intensity per area in each aquarium is ensured since its distance to the tank is always constant. Moreover, it can be seen in both Figures 4.4 a) and b), that there is specific hardware selected for both video recording and user interaction.

In this way, to record the videos, a Raspberry Pi 2, Model B [19] is used with an integrated camera [20] (Figure 4.7). On the front side of the container there is a fixed touch screen [21] that transforms the container into an interactive setup for the user, representing an all-in-one recording system. As the backside of the screen is connected to the Raspberry Pi and camera system, as it can also be seen in Figure 4.7, it guarantees that the camera is always in the same place and, again, the distance between the tank and the camera remains constant. Finally, Figures 4.8 and 4.9 show nacre adults and fries respectively, examples of resulting images obtained by the latest version of the Danio Recording Setup.

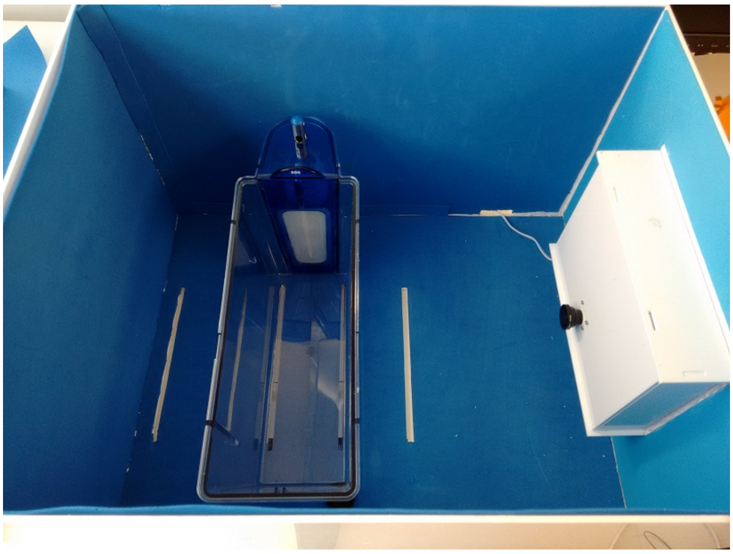


Figure 4.6: Empty tank placed inside the Danio Recording Setup.

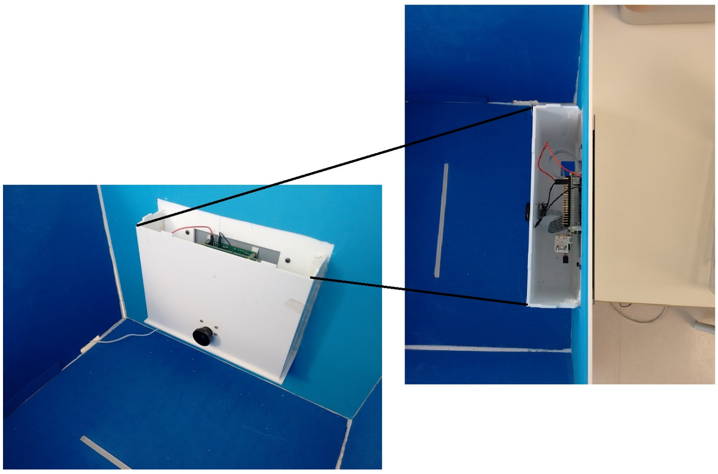


Figure 4.7: Raspberry Pi and camera housing.

4.2 Estimating the Number of the Fish

After video recording with the recording setup, the task of image processing begins. The proposed method, that we next describe in detail, was applied to every frame in a given video and can be generally summarized in the following steps:

- Background substraction with GMMs and noise removal
- Blob detection and fish count based on pixel area
- Shoaling checking
- Mirroring compensation using Hu moments

The summarized solution can be found on the pseudocode in subsection 4.2.7 where all steps for a video with, with no shoaling, are considered.

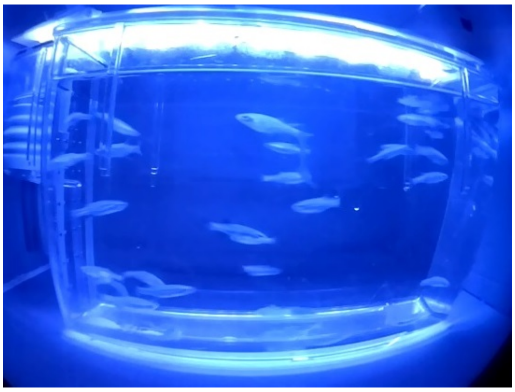


Figure 4.8: Frame obtained by the latest Danio Recording Setup version for 27 Nacre adult fish.

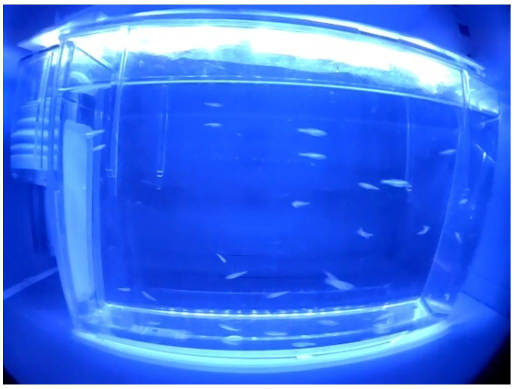


Figure 4.9: Frame obtained by the latest Danio Recording Setup version for 27 Nacre fries.

It is also important to state that this solution was implemented using Python programming language as well as OpenCV Library, numpy and matplotlib.

4.2.1 Background Subtraction with Gaussian Mixture Models

Regarding video properties, it is relevant to state that each frame has a size of 640x480, resulting in an image as shown in Figure 4.8, and is acquired at a frame rate of 25 Hz. Each frame is initially cropped, obtaining a 490x300 image, so that the tank can be the only region of interest in it, Figure 4.10, reducing the processing time. Then, the algorithm of background subtraction described in section 3.1, is applied to the cropped frame with the tank.

As previously described, the background subtraction algorithm uses a GMM background subtraction scheme [11] [13] that automatically selects specific components, for each pixel in the image. This makes the differentiation between what is considered as background (blue environment) and foreground (fish moving between successive frames) possible. This technique allows adaptability during frame variation to guarantee that the background is consistently subtracted and only foreground variations are able to be identified. The result, Figure 4.11, is a black and white image, where black represents the background and white the fish (blobs, from now on).

4.2.2 Noise Filtering

We can see in Figure 4.11 that there is noise (very small white regions) present in the frame and the areas representing the fish are not closed ones. At this point, a routine of blob dilation is applied resulting in Figure 4.12. In this image the blobs representing the fish are now closed, however, the noise regions became larger. To avoid the noise influence in counting, given the image on Figure 4.12, we choose only the blobs with area above 100 pixel since, in this case, we guarantee that everything in the frame is most likely to be fish.

The result of this region selection can be found in Figure 4.13 that represents the final stage from which the counting starts to be done. It is important to state that we could have applied an erosion routine

Fish Category	Reference area [pixel]
AB/TU Fries	150
Nacre Fries	250
AB/TU Adults	500
Nacre Adults	600

Table 4.1: Reference area for one fish in the different categories.

at Figure 4.11 to eliminate the noise and only then apply the dilation, however, this would represent more processing time.

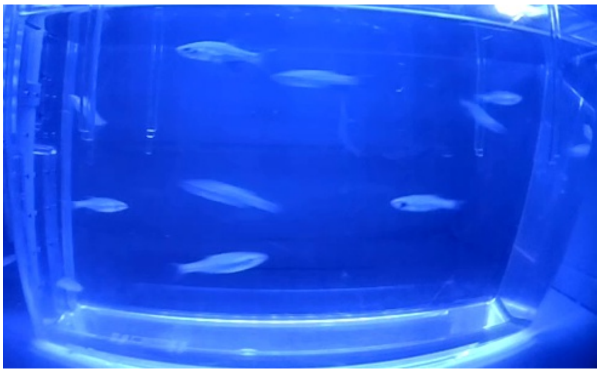


Figure 4.10: Cropped frame.

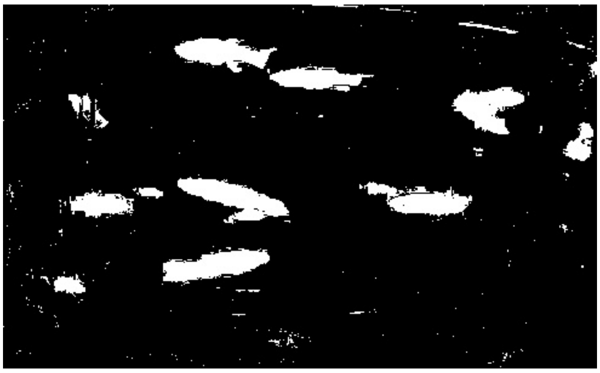


Figure 4.11: Cropped frame after background subtraction.

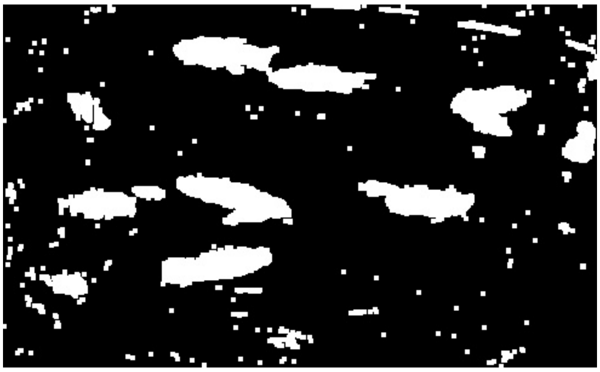


Figure 4.12: Frame after dilation

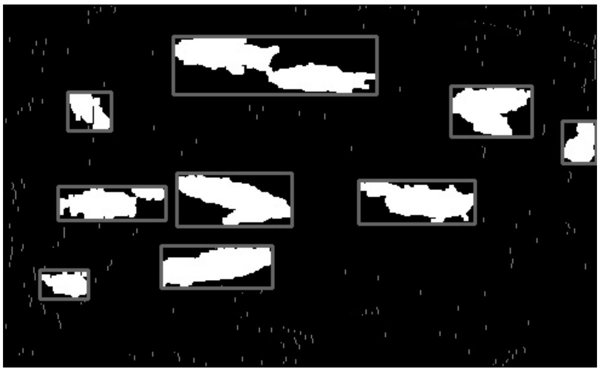


Figure 4.13: Frame ready to be analysed.

4.2.3 Heuristic Blob Counting

After dilation, a routine for blob contours detection takes place. To define the threshold from which a blob is considered to represent a fish or multiple fish, video testing was previously performed. In this testing, 5-minute videos (7500 frames) containing only one fish were recorded to obtain the average area (which will be called reference area from now on) that a fish assumes in the video. The approximated results are shown in Table 4.1.

With the contours detection routine, we can identify (grey boxes around the blobs in Figure 4.13) and count blobs larger than 100 pixel (too exclude the noise as previously stated) and count them to obtain a first estimate of the fish in one frame. Nevertheless, fish overlap resulting in blobs significantly bigger than the reference area. In this project we consider that overlapping occurs every time at least two fish are close enough to form one single blob as in Figure 4.14 for two Nacre adult fish.

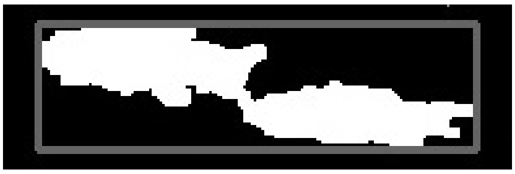


Figure 4.14: Two Nacre adult fish overlapping.

Visually, both fish in this frame have approximately the same area and are represented in only one blob. Lets assume that each of these fish has an area approximately equal to the reference area. In this case the area of the blob would be two times the reference area, thus, dividing this value for the reference area we get the number 2. Hence, where we previously counted this blob as one fish, we now have a correction for this value representing the correct number of fish in this blob. This procedure is done for every blob with area above the reference area for each category. Different reference areas than the calculated in the tests are applied as the number of fish increases in the tank. In fact, after testing in ten 40-second videos for each category and school of fish (different from the ones from where results were extracted), it was possible to build intervals, as seen from Table 4.2 to Table 4.5, with typical areas that specific quantities of fish represent. This way, if a frame has a total blob area (sum of the area of all the blobs in a frame) in one of this intervals, a specific reference area is used to compensate fish overlapping.

The reference areas that were used to compensate the counting of overlapping fish in frames can be found from Table 4.6 to Table 4.9. It is important to refer that in the first interval (the one with lesser number of fish), the reference areas that are used are greater than twice the values at Table 4.1 since we admit that the number of fish in the tank is low enough to have a sufficient number of frames without overlapping. In this way, the compensation does not need to be done as intensively as in cases with more fish. We did also assume that the second intervals represent the typical number of fish where overlapping compensation needs to start being done more frequently. Thus, the second intervals have twice the reference areas in Table 4.1.

It is understandable that, for instance, if the number of fish is 35, a blob with the same size as a case where there are 25 fish in the tank may represent more fish. This is the reason why as the number of fish increases in the fish intervals, we decrease the value of the reference area used by 100 pixels (starting at the second interval) so when we divide an overlapping blob area by the reference area the compensation is greater.

To provide better understanding of the use of the values in these tables, we will describe a practical example of a tank containing 20 AB/TU adult fish. Initially, there is no information of how many fish are inside a tank thus, the algorithm calculates, for each frame, the total blob area. In cases were shoaling does not occur, the total blob area in a frame will be in the interval present in Table 4.4, $]9000, 17000]$. Having obtained a number in this interval, we then automatically know that the reference area to be used for fish overlapping compensation in this frame is, from Table 4.8, 1000 pixel.

AB/TU Fries	
Fish Number Interval	**Typical area [pixel]**
$[5, 17]$	≤ 4500
$[20, 22]$	$]4500, 5200]$
$[25, 35]$	≥ 5200

Table 4.2: Typical areas for AB/TU fries for the different sizes of the fish school.

Nacre Fries	
Fish Number Interval	**Typical area [pixel]**
$[5, 12]$	≤ 3000
$[15, 25]$	$]3000, 5000]$
$[27, 35]$	≥ 5000

Table 4.3: Typical areas for Nacre fries for the different sizes of the fish school.

AB/TU Adults	
Fish Number Interval	**Typical area [pixel]**
$[5, 12]$	≤ 9000
$[15, 25]$	$]9000, 17000]$
$[27, 30]$	$]17000, 20000]$
$[32, 35]$	≥ 20000

Table 4.4: Typical areas for AB/TU adults for the different sizes of the fish school.

Nacre Adults	
Fish Number Interval	**Typical area [pixel]**
$[5, 10]$	≤ 9000
$[12, 22]$	$]9000, 17000]$
$[25, 27]$	$]17000, 20000]$
$[30, 35]$	≥ 20000

Table 4.5: Typical areas for Nacre adults for the different sizes of the fish school.

AB/TU Fries	
Fish Number Interval	**Reference area [pixel]**
$[5, 17]$	400
$[20, 22]$	300
$[25, 35]$	200

Table 4.6: Reference area used for AB/TU fries for the different sizes of the fish school.

Nacre Fries	
Fish Number Interval	**Reference area [pixel]**
$[5, 12]$	600
$[15, 25]$	500
$[27, 35]$	300

Table 4.7: Reference area used for Nacre fries for the different sizes of the fish school.

AB/TU Adults	
Fish Number Interval	**Reference area [pixel]**
$[5, 12]$	1500
$[15, 25]$	1000
$[27, 30]$	900
$[32, 35]$	800

Table 4.8: Reference area used for AB/TU adults for the different sizes of the fish school.

Nacre Adults	
Fish Number Interval	**Reference area [pixel]**
$[5, 10]$	1500
$[12, 22]$	1000
$[25, 27]$	900
$[30, 35]$	800

Table 4.9: Reference area used for Nacre adults for the different sizes of the fish school.

4.2.4 Shoaling Behaviour in Fish

During the development of this project, we could conclude that zebrafish group's motion tend to vary between the following "modes": randomly spread across the tanks, schooling and shoaling. Fish are considered to be shoaling if they stay together for social reasons and, in cases where they move coordinated in the same direction, their behaviour is classified as schooling. Zebrafish schools are faster and less dense than zebrafish shoals. The habituation to a new environment can alter the proportion of time zebrafish groups spend schooling or shoaling [22].
In Figure 4.15 we can see an example of 22 Nacre adult fish shoaling at the bottom of a tank. For each frame our algorithm is capable of identifying shoaling at the bottom of the tank and at the at the sides of the tank. Since in our samples every shoaling that occured was on the bottom of the tank, the figures on this report illustrate that specific situation.

In order to identify shoaling, after blob counting a frame as for instance the one in Figure 4.15 is splitted. We can see the two resulting splitted parts of the frame in Figures 4.18 and 4.19. At this point

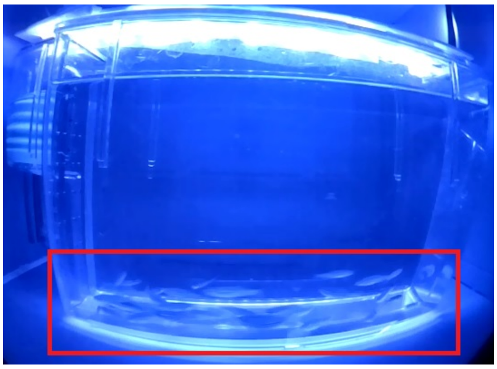


Figure 4.15: 22 Nacre adult fish shoaling at the bottom of a tank.

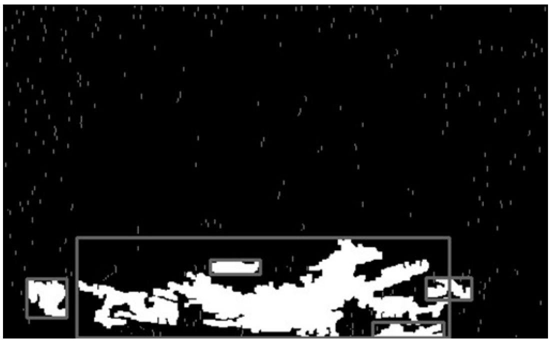


Figure 4.16: Frame resulting from the analysis of the 22 Nacre adult fish shoaling at the bottom of the tank.

we verify if 90% of the total blob area detected in a frame is present at the split part corresponding to the bottom of the frame. In cases this is verified we are then able to say that shoaling is occuring. As we can see in Figure 4.19, only 5 blobs are identified by the algorithm when, in fact, there are 35 Nacre adult fish inside the tank. It is understandable that when this type of situation occurs is very difficult for the algorithm to make fish overlapping compensation by dividing each blob area by the reference area since there are multiple layers of fish behind one another. These images are 2D so there is no information regarding the depth of the shoal to relate to this frames and implement other compensations. In these situations the total blob area that is detected by the algorithm, in this case for 35 Nacre adult fish, will be, for instance, inside the interval $[5, 12]$ when in a normal situation with fish spread across the tank, the area would be in the interval $[30, 35]$. In this way, the reference area to compensate overlapping would be, from Table 4.8, 1500 pixel instead of 800 pixel. Thus, we can conclude at this point that the frequent

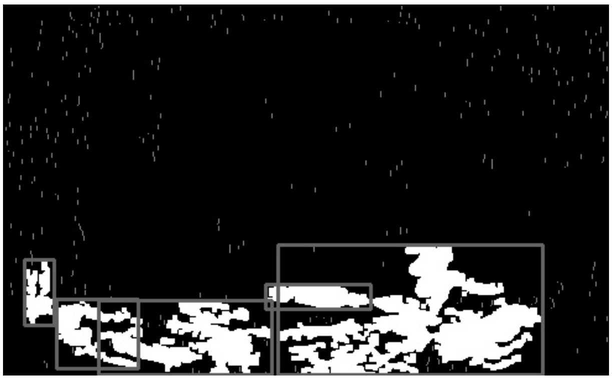


Figure 4.17: Frame before splitting and shoaling checking with 35 AB/TU adult fish.

Figure 4.18: Resulting top part of the frame splitting.

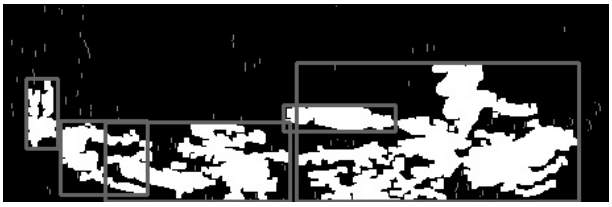


Figure 4.19: Resulting bottom part of the frame splitting with 35 AB/TU adult fish.

presence of shoaling in frames may lead to unacceptable results if a significant number of frames verify its existence.

In this project when frames with shoaling are detected, the counting that is obtained in those frames is used for the final estimation of the number of fish. We decided to do it (we could have ignored those frames) just to reflect how shoaling affects the algorithm performance.

4.2.5 Mirrored Counting Compensation

Mirroring compensation is done by calculating each Hu moment for each blob in a frame based on Ming-Kuei Hu [17] work that presents a theory of two-dimensional moment invariants for planar geometric figures.
In the recorded videos two areas containing fish mirroring can be detected: the right side of the tank and the upper part (water surface). Near these regions fish are reflected like in a mirror which may lead to count some fish twice. We can see an example of a fish mirroring at the surface of the tank in Figures 4.20 and 4.21 with the two blobs inside the red rectangles.

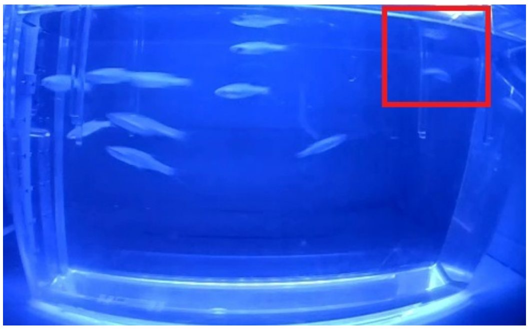


Figure 4.20: One fish mirroring at the top of the tank.

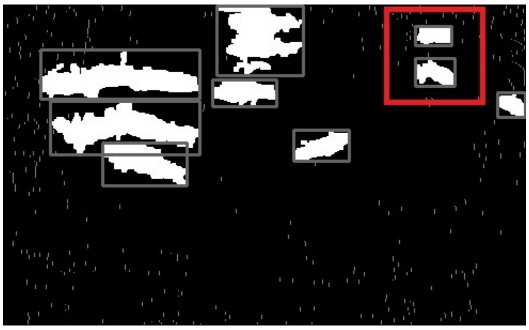


Figure 4.21: Frame to analyse with one fish mirroring at the top of the tank.

Given a frame the algorithm calculates, for each blob, its Hu moments. After analysing frames where we could visually identify mirroring, we studied how the values obtained for the orthogonal invariants described in Chapter 3, Section 3.3 behaved. The first orthogonal invariant gives information regarding shape features while the seventh invariant allows the identification of mirroring [17]. In fact, after analysing frames where reflections occurred, we were able to conclude that the blob that represented the real fish and the blob belonging to the mirroring had approximately the same value for the first orthogonal invariant (differences $\leq 5 \times 10^{-3}$) and the seventh invariant had approximate absolute value and opposite signal. We can have a look at these values in Table 4.10. Thus, after calculating every Hu

	Blob representing fish	**Blob representing the mirroring**
First orthogonal invariant	0.589409646739	0.592189044417
Seventh orthogonal invariant	8.44891493114	-8.00010965607

Table 4.10: Example of the first and seventh orthogonal values for a blob representing a fish and another representing the mirroring in Figures 4.20 and 4.21.

moment for each blob in one frame, we compare the first and the seventh orthogonal invariants of one blob to the other blob's invariants. Hence, every time two blobs in a frame verify:

- the difference between their first orthogonal invariants is $\leq 5 \times 10^{-3}$
- the difference between their seventh orthogonal invariants absolute values is ≤ 1 and have opposite signal

we know that the number of reflections detected should be subtracted to the counting done so far. This correction is the last modification that the algorithm applies to the counting in one frame. After this step we get the final estimate for one frame and the process repeats itself from the beginning, subsection 4.2.1, until this point.

4.2.6 Outputting the Final Number of Fish in a Video

Having completed all the steps, for each frame of a video, described from subsection 4.2.1 to 4.2.5, we can provide an estimate for the number of fish in a tank.

Let us denote the estimated number of fish detected in each frame of a video by: f_m^n, $n = 1, ..., N$ and $m = 1, ..., M$, where N and M are the number of videos and number of frames per video, respectively. The final number of fish in a video, n, is finally given by: $\hat{F}_n = median(f_1^n, f_2^n, ..., f_M^n)$.

Consequently, the error in a video can be calculated:

$$e_n = \frac{|\hat{F}_n - K_n|}{K_n} \tag{4.1}$$

where K_n represents the real number of fish, which is known *a priori*. Finally, we can compute the average error for N videos:

$$\bar{e} = \frac{\Sigma_{n=1}^{N} e_n}{N}. \tag{4.2}$$

In this project we decided to use the median instead of, for instance, the mean. This was made taking into consideration that in certain frames (particularly in the first frames of the video where the background subtraction algorithm is still stabilizing) the count estimate is significantly higher (or lower in partial shoaling cases) than the real number of fish. Hence, since median in opposition to mean is less susceptible to outliers, this measure was chosen. With this outliers filtering, it is possible to guarantee that, after a certain number of frames, the algorithm tends to converge for a specific counting value.

4.2.7 Solution Summary

At this point we are able to summarize the different steps of the developed solution through the following pseudocode:

```
Data: Video sample
Result: Fish number estimate
load video sample
while there are frames to read do
    apply background subtraction to current frame
    apply dilation to current frame
    find blob contours greater than pixel threshold
    perform blob counting to get a first estimation
    if fish category is AB/TU fries then
        reference area = AB/TU fry average area
    end
    else if fish category is Nacre fries then
        reference area = Nacre fry average area
    else if fish category is AB/TU adults then
        reference area = AB/TU adult average area
    else if fish category is Nacre adults then
        reference area = Nacre adult average area
    if there are blob areas larger than reference area then
        for each blob area do
            blob compensation = blob area $\div$ reference area - 1
            first compensation = blob compensation + first compensation
        end
        second counting = first estimation + first compensation
    end
    for each blob do
        calculate Hu moment (orthogonal invariants) for mirroring compensation
        if other blobs have first and seventh invariants that verify: difference between first
        invariants $\leq 5 \times 10^{-3}$ and difference between seventh invariants absolute values is $\leq 1$ and
        have opposite signal then
            mirroring compensation = number of times 'if' condition is verified
        end
    end
    fish number estimate in frame = second counting - mirroring compensation
    estimates in sample.append(fish number estimate in frame)
end
final fish number estimate = median(estimates in sample)
```

Algorithm 1: Fish counting estimate - procedure summary.

In the previous algorithm summary, there is no information regarding shoaling detection since we decided to include shoaling when calculating the estimates to be able to analyse the effect of this behaviour in the final results. However, we check if the shoaling occurs by analysing if 90% of the detected blobs are in a specific region of the tank as previously explained and if we included this procedure in Algorithm 1 it would be after calculation the Hu moments for each blob.

4.2.8 A Note on Optical Flow

The optical flow technique [3] is based on a two-frame motion estimation algorithm as described in Chapter 3, Section 3.2. In this method neighbourhoods of both frames are approximated by quadratic polynomials using the polynomial expansion transform. In the end, displacement fields are derived from polynomial expansion coefficients by observing the way an exact polynomial transforms under translation.

Taking this into consideration, an approach using optical flow was initially used after background subtraction. That was thought since each blob's displacement vector features gave the possibility to differentiate between fish overlapping in case they where moving in opposite directions.

The output of the optical flow algorithm applied to one frame consisted on the calculation of a displacement vector, for each different blob. This displacement vector is described by a magnitude and an angle that represents direction. Consequently, if each blob moves in different directions, each blob's displacement vector is represented by distinct values of magnitude and angle. In order to have a visual representation of this values for each blob, we convert the cartesian values of the magnitude and angle to polar coordinates, Figure 4.22, in order to express each combination of values as HSV (Hue, Saturation and Value) colors. We can see in Figures 4.23 and 4.24 how HSV color scale varies. Thus, after the previously stated conversion we get a color for each pair $(magnitude, angle)$. We can see in Figure

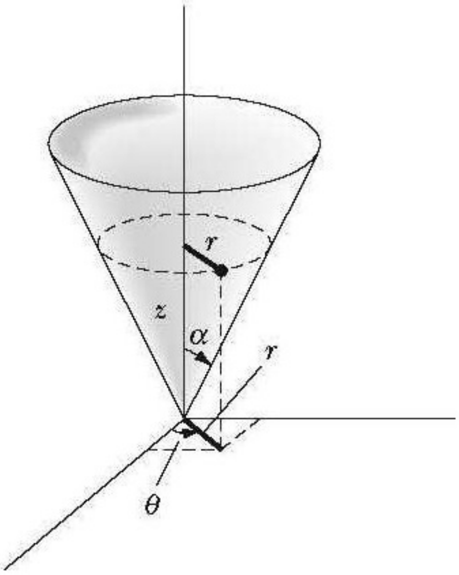


Figure 4.22: Polar coordinates. [23]

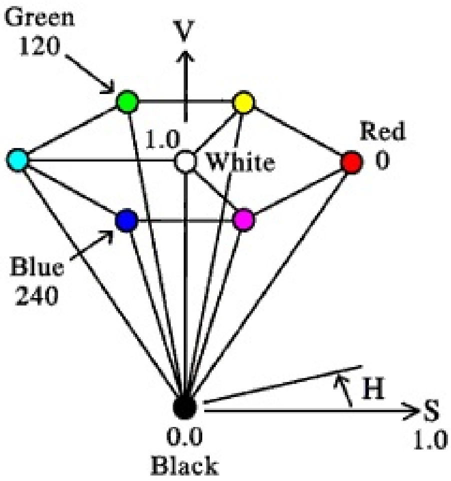


Figure 4.23: HSV color distribution. [24]

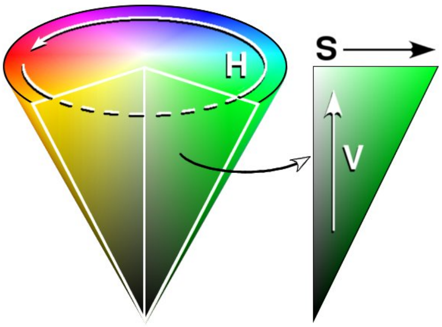


Figure 4.24: HSV color cone. [25]

4.26 the visual representation of the displacement vector for every blob detected in the original frame, Figure 4.25.

After obtaining the frame with the optical flow representation, pre-built color intervals were used to check which colors fit in these intervals and can be found in Table 4.11. In this way, we were able to analyse, separately, blobs with the same color (Figure 4.27 to Figure 4.36). This made possible to differentiate between fish overlapping in cases where fish moved in different directions. Then, even when crossing in front of each other, fish' blobs would exhibit different colors. Finally, after splitting all the colors, we could count how many blobs were in each of the five resulting image and sum them. The result represents the estimate for a specific frame.

	Hue	Saturation	Value
Reds	$[0, 20]$	$[0, 255]$	$[0, 255]$
Yellows	$[21, 35]$	$[0, 255]$	$[0, 255]$
Greens	$[36, 92]$	$[0, 255]$	$[0, 255]$
Blues	$[93, 130]$	$[0, 255]$	$[0, 255]$
Purples	$[131, 180]$	$[0, 255]$	$[0, 255]$

Table 4.11: HSV defined color intervals.

However, after testing, we realized that when there was a significant number of frames where fish moved with very low speed, it was impossible to detect them with this method. Processing time would also increase comparing to the final solution that was previously described since optical flow computation implicates additional calculations. These were the reasons why optical flow was not further considered in this algorithm and the final solution remained with the application of background subtraction and the area compensation as described in subsection 4.2.1.

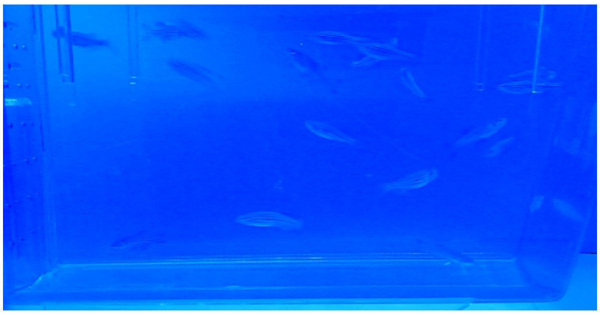


Figure 4.25: Original frame to apply background subtraction and then calculate the optical flow.

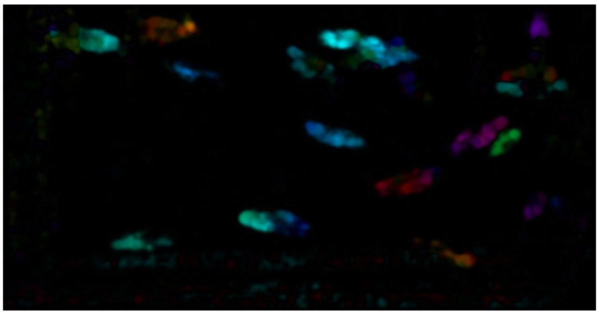


Figure 4.26: Representation of the optical flow for each blob in the original frame.

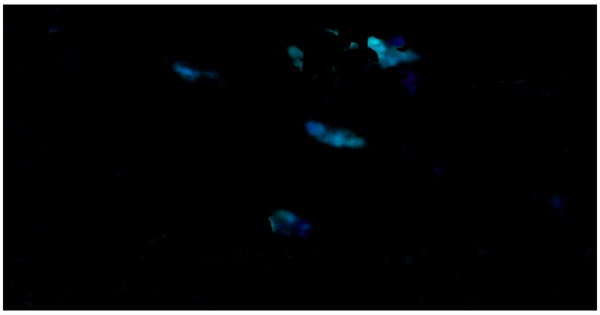


Figure 4.27: Separation of the blobs with blue colors from the optical flow representation.

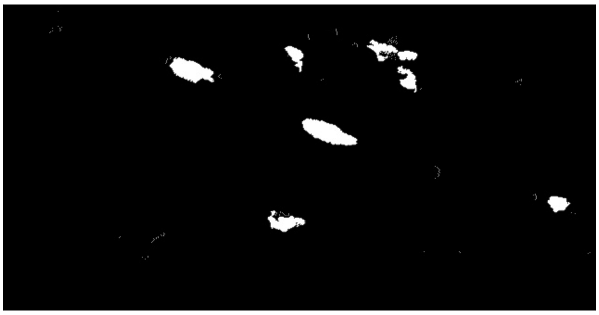


Figure 4.28: Blobs representing blue regions.

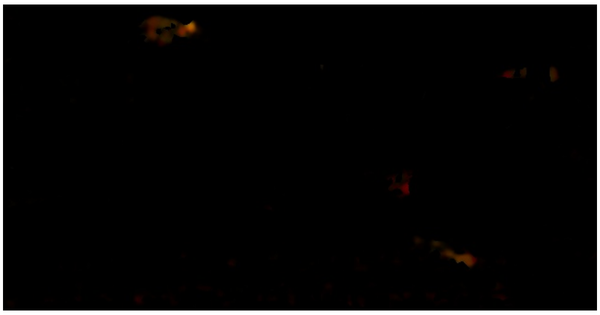


Figure 4.29: Separation of the blobs with red colors from the optical flow representation.

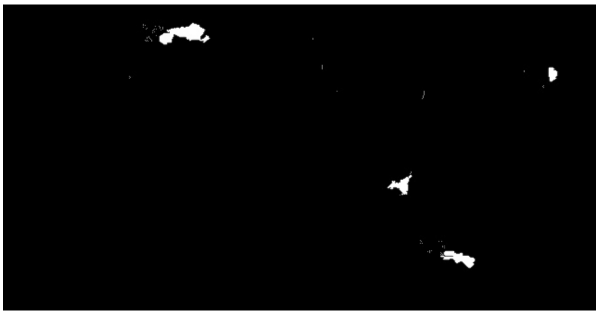


Figure 4.30: Blobs representing red regions.

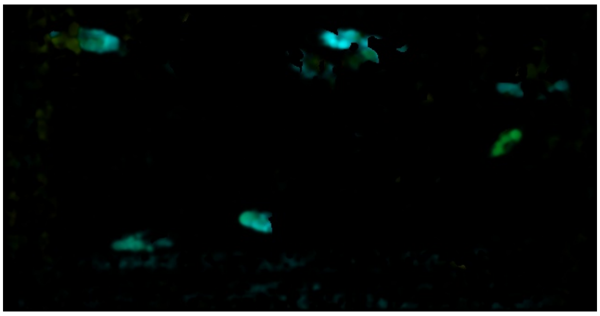


Figure 4.31: Separation of the blobs with green colors from the optical flow representation.

Figure 4.32: Blobs representing green regions.

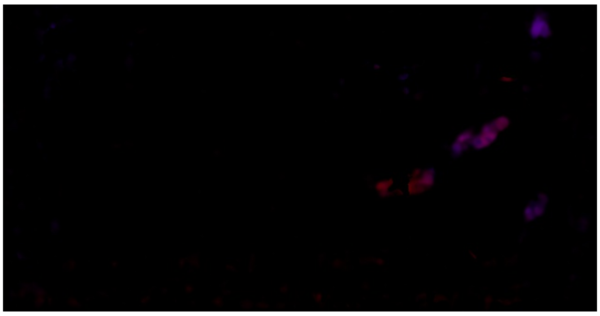


Figure 4.33: Separation of the blobs with purple colors from the optical flow representation.

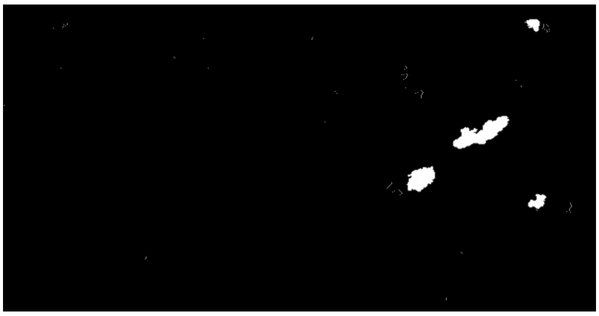


Figure 4.34: Blobs representing purple regions.

Figure 4.35: Separation of the blobs with yellow colors from the optical flow representation.

Figure 4.36: Blobs representing yellow regions.

Chapter 5

Evaluation

To begin with, it is important to state that videos with fish were recorded at Champalimaud Centre for the Unknown Fish Facility and were processed offline. Fish were raised and maintained at the Champalimaud Fish Platform according to Martins et al. (2016) and manipulated by staff accredited for animal experimentation by the Portuguese Veterinary Agency (DGAV)[26].

In this implementation we considered the following number of fish in a school for each of the four categories of genotype vs age: 5, 7, 10, 12, 15, 17, 20, 22, 25, 27, 30, 32, 35 representing a total of 13 categories. The only exception is AB/TU fries which has 10 categories since there were no tanks with more than 27 30-day old zebrafish by the time recordings were made.

For each different category and school size we collected 20 videos of 40 seconds each, thus totalling 200 videos for AB/TU fries and 260 for each of the remaining categories.

5.1 Results - Final Solution

After applying the described counting algorithm to those 40 second videos, we obtained the counts in different 20 trials that can be found in Tables A.1, A.3, A.5 and A.7 in Appendix A. The average error resulting from the counts can be found in Table 5.1 where acceptable average errors ($\leq 15\%$) are indicated in green and not acceptable in red.

Let us recall that the estimated number of fish detected in each frame of a video is denoted by: f_m^n, $n = 1, ..., N$ and $m = 1, ..., M$, where N and M are the number of videos and number of frames per video, respectively. The number of fish is given by: $\hat{F}_n = median(f_1^n, f_2^n, ..., f_M^n)$. The error in a single video is expressed by equation 4.1 and the average error for N videos is calculated using 4.2 where in this case $N = 20$ for each entry of Table 5.1.

Regarding fries' category, we can see that the highest error observed was 15%, for Nacre fries in a 10 fish tank. For the remaining schools of fish acceptable errors were obtained.

Since fries are significantly smaller than adults, as seen in Figures 4.8 and 4.9 and previously stated in Section 1.2, overlapping is less frequent and cases where the error is higher (than 15%) correspond to partial shoaling. Partial shoaling in a video sample may be defined as the occurrence of shoaling only

	Fish Number												
Fish Category	**5**	**7**	**10**	**12**	**15**	**17**	**20**	**22**	**25**	**27**	**30**	**32**	**35**
AB/TU Fries	4%	13%	7%	10%	4%	9%	7%	9%	4%	3%	■	■	■
Nacre Fries	14%	12%	15%	6%	8%	6%	11%	3%	4%	5%	1%	8%	4%
AB/TU Adults	0%	6%	5%	3%	9%	11%	14%	3%	5%	24%	15%	12%	29%
Nacre Adults	21%	13%	18%	12%	29%	6%	9%	20%	9%	11%	7%	3%	4%

Table 5.1: Average fish count error for each number of fish in the different categories (20 samples per category) obtained in the final solution.

in portions of a video in such way that there are enough frames where shoaling is not detected leading to an estimate in the acceptable margin.

5.1.1 Shoaling

In the adult fish category, there were certain fish quantities which did not meet the acceptable margin. For instance, in the videos with 15 Nacre adult fish the average error is 29% which means that, in average, approximately 5 fish were not detected in those videos which is far from the real value. This is due to the significant high number of videos where shoaling occurred. As previously mentioned, fish shoal particularly in the bottom of the tank and it is extremely difficult for the algorithm to output correct counts when this behaviour is verified. This occurs for videos with few or many fish when fish are closely together in multiple layers behind each other. Hence, shoaling justifies the average errors at Table 5.1 that are higher than the 15% margin.

In Figure 5.1 representing 22 adult fish, we can see that for videos where shoaling occurs, the percentage of total blob area detected in each frame is significantly lower than the case in Figure 5.2 where shoaling does not occur. Thus, it is understandable that if there is less blob area than it should in a frame, it will lead to unacceptable results.

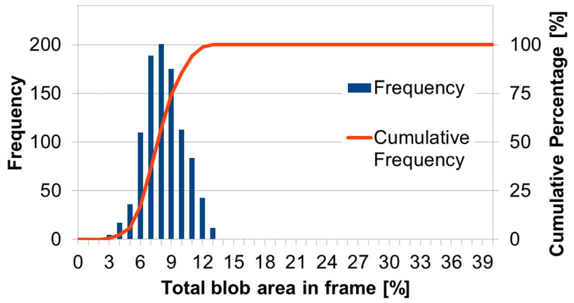


Figure 5.1: Frequency and cumulative frequency histogram for a video with 22 fish shoaling.

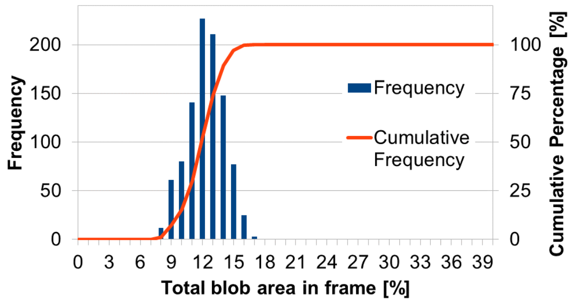


Figure 5.2: Frequency and cumulative frequency histogram for a video with 22 fish without shoaling.

5.1.2 Algorithm Convergence

We can see examples of how the fish count per frame varies within individual videos for fish in the different categories from Figures 5.3 to 5.6.

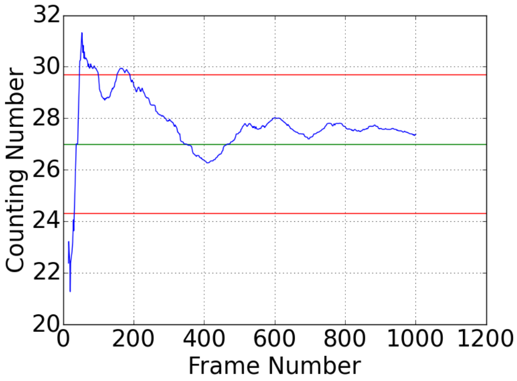


Figure 5.3: Counting number for each frame for 27 AB/TU zebrafish fries in a 40-second video (1000 frames). The red lines represent the acceptable error margin and the green line is the real fish number in the tank).

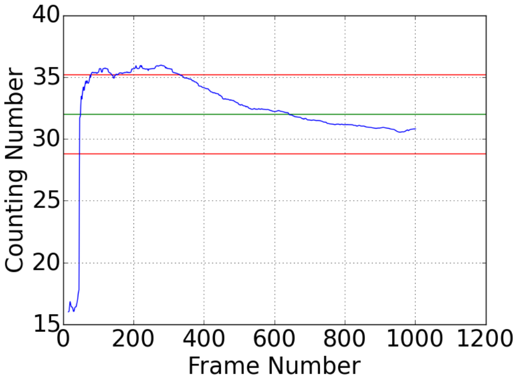


Figure 5.4: Counting number for each frame for 32 adult AB/TU zebrafish in a 40-second video (1000 frames).

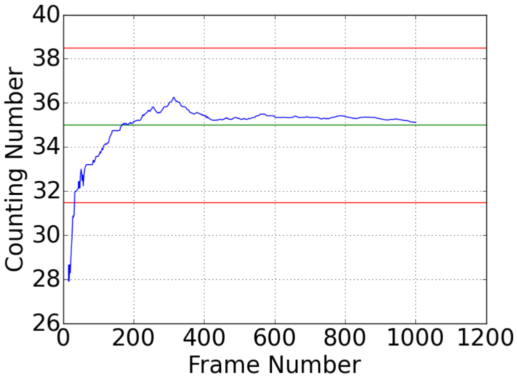


Figure 5.5: Counting number for each frame for 35 Nacre zebrafish fries in a 40-second video (1000 frames).

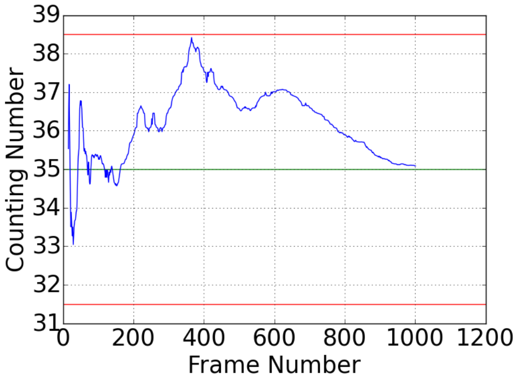


Figure 5.6: Counting number for each frame for 35 adult Nacre zebrafish in a 40-second video (1000 frames).

In those figures, the real fish value inside the tank is represented by the green line and $\pm 15\%$ of the real fish number correspond to the red lines. It can be verified that the algorithm outputs a number within the error margin in early frames and maintains the error margin until 1000 frames. Nacre zebrafish tend to reflect more the light, resulting in bigger blobs for single fish, comparing to the AB/TU category, since these are darker, which explains the multiple peaks in Figure 5.6. This peaks are not so evident in Nacre fries, Figure 5.5, due to their small dimensions which, in the ends, compensates the reflection.

From Figure 5.7 to 5.10 we can analyse the variation of the variance for each school of fish in the different categories (this information can be found in Tables A.2, A.4, A.6 and A.8 in the Appendix A as well as graphs with the standard deviation from Figure A.1 to A.4). Regarding Nacre, Figure 5.7

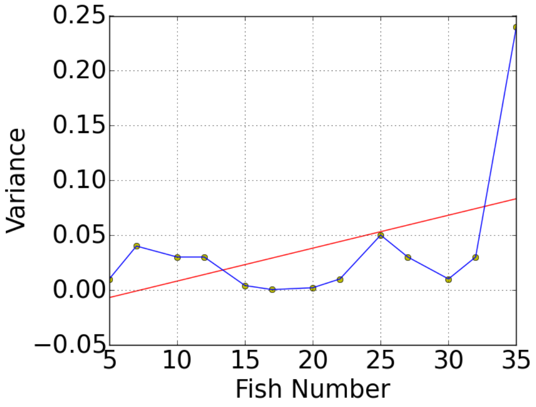


Figure 5.7: Plotted variance values and linear regression for each school of Nacre fries in Table A.2.

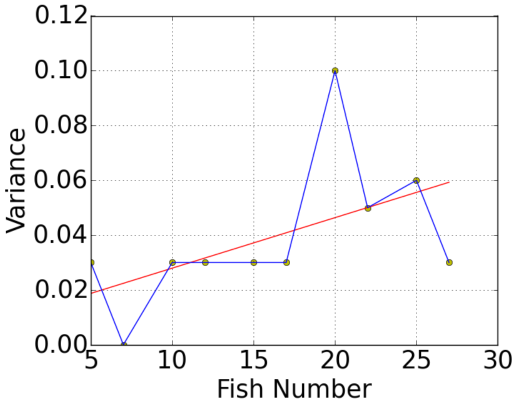


Figure 5.8: Plotted variance values and linear regression for each school of AB/TU fries in Table A.4.

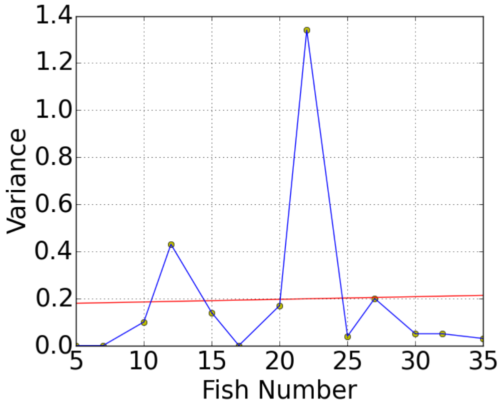


Figure 5.9: Plotted variance values and linear regression for each school of Nacre adults in Table A.6.

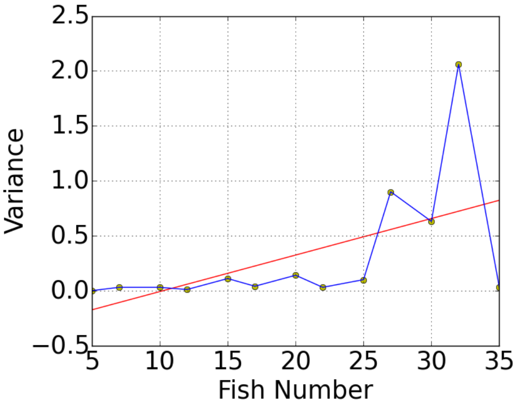


Figure 5.10: Plotted variance values and linear regression for each school of AB/TU adults in Table A.8.

and AB/TU fries, Figure 5.8, categories, we can easily verify that the variance values are very low. This means that the algorithm presented results which were very close to the average error in the 20 different video samples for each school number. Taking into consideration that all the values in Table 5.1 for fries are within the acceptable margin and the error variances are very low, we can conclude that the algorithm performs well in this category. Moreover, if we analyse the regression lines in red, we can verify that the line slope is positive. This demonstrates that the error's variance tends to increase with the increase of the number of fish inside a tank as expected.

In Nacre and AB/TU adult categories, there are some error's variance values that are significantly greater than the ones in fries categories. This is due to the fact that adult fish have bigger size which makes overlapping occur more often leading more frequently to situations where shoaling and partial shoaling occur. Hence, in some trials (27 AB/TU adult fish, for instance), these behaviours were more intense than in others leading, for example, to high error values or even no error in some video samples. These variations lead to the increase of the variance value. There were also cases in which shoaling occurred similarly in all the video samples, as in 35 Nacre adult category, leading to an average error very close to the (wrong) counting values obtaining, consequently, low variance. It was once more possible, through the analysis of the regression lines, to verify that the slope is positive representing the same tendency observed in fries categories (variance tends to increase with the increase of the number of fish inside the tank).

5.2 Mirroring Compensation

In order to emphasize the importance of the fish mirroring compensation (i.e., subtracting blobs representing fish reflections that were previously counted as fish), we present, in this section, the results of the algorithm, over the same samples as in the previous section, without the calculation of Hu moments (mirroring identification). The counting results may be found in Table 5.2. When performing fish counting with the final version of the algorithm, we can obtain an estimate of the counting equal, above or below the real number of fish inside the tank. If the estimate, in a frame, is lower than the real value of fish, the overall counting estimate value tends to decrease and the error of the estimate increases when we compensate the blobs that represent fish mirroring. On the other hand, if before mirroring compensation the estimate of the counting is greater than the real value, when we subtract the mirroring counts, the value of the estimate tends to decrease and converge to the real number of fish inside the tank. Having this into consideration, we can see in Table 5.2 how the error varies, when comparing to Table 5.1, if the mirroring compensation was not performed. In fact, we can easily verify that in Table 5.2 we have many more situations where the 15% error margin was not achieved. For instance, for 20 AB/TU adults we obtained 14% error margin in Table 5.1, when mirroring compensation is done and in Table 5.2 we can see that the error increases to 21% representing a value outside the margin.

	Fish Number												
Fish Category	**5**	**7**	**10**	**12**	**15**	**17**	**20**	**22**	**25**	**27**	**30**	**32**	**35**
AB/TU Fries	4%	16%	9%	16%	11%	12%	18%	13%	9%	7%	■	■	■
Nacre Fries	14%	11%	17%	8%	6%	11%	13%	7%	10%	12%	9%	12%	5%
AB/TU Adults	0%	13%	17%	10%	12%	18%	21%	9%	11%	24%	17%	13%	29%
Nacre Adults	21%	13%	18%	17%	29%	18%	16%	20%	13%	12%	15%	9%	7%

Table 5.2: Average fish count error for each number of fish in the different categories without mirroring compensation (20 samples per category).

We can also verify that, for instance in 7 and 15 Nacre fries, the error margin is lower in Table 5.2. This means that before compensation, the count estimate was below the real number of fish. Only in this situation can the average error in Table 5.2 be lower than in Table 5.1.

It is also important to verify that the cases which we identified as shoaling in Table 5.1, for instance 27 AB/TU adults and 22 Nacre adults, present the same value in both tables. This is due to the fact that the blobs that represent the shoaling occurred mostly at the bottom of the tank where no reflections are detected. Moreover, since blobs that results from shoaling present, generally, very distinct shape characteristics among each other, the first condition in Hu moments comparison (shape features in the first orthogonal invariant) is not verified leading to a situation where reflections can not be identified.

5.3 Algorithm Time Performance

In this section we present examples of execution time of the algorithm in videos with different quantities of zebrafish. It is possible to find those values in Table 5.3 where times for adults, Table 5.3 a) and fries, Table 5.3 b), are represented.

Fish Number	Execution Time[s]
5	18,2
7	26,4
10	25,5
12	24,4
15	25,2
17	29,5
20	32,4
22	31,9
25	36,6
27	37,8
30	38,1
32	41,3
35	44,9

a)

Fish Number	Execution Time[s]
5	14,1
7	15,2
10	17,4
12	17,7
15	17,9
17	19,5
20	23,3
22	23,7
25	24,7
27	28,6
30	29,3
32	28,1
35	28,7

b)

Table 5.3: Examples of the amount of time (in seconds) to run the algorithm in videos with different number of Nacre zebrafish adults, a) and fries, b).

It is important to state that this examples were obtained in a single 40-second video, for each number of fish, using an Intel Core i7-4500U CPU @ 1.8GHz - 2.4GHz processor and 8GB of RAM.

We can easily verify that this offline video processing never takes more than approximately 45 seconds. That value is verified in the (Nacre) adults category, as expected, since adult fish are represented by bigger blobs which demands more time to process.

It is interesting to compare these values to the ones present in Table 1.1 where manual counting times for zebrafish fries are presented. Fries' counting is more time consuming comparing to adults counting due to the fish' size. If we compare the two tables, we can see that in Table 1.1 most of the counting times exceed or is approximately equal to 30 seconds and the decrease in the number of fish does not always traduce the decrease in the amount of time. Using the information in both tables, we can, finally, conclude that the time our algorithm takes to run a 40-second video (1000 frames) is acceptable taking into consideration the manual counting time for fries. Even though adult fish tend to be easier to count manually than fries, we can also consider the execution time for adults as acceptable since the longer our algorithm takes to output an estimate is around 45 seconds for 35 fish (only 15 seconds more than the maximum obtained for the fries).

Other processing was made using an Atom where a 40-second video took approximately 80 seconds to be analysed and a Raspberry Pi outputted an estimate after approximately 2 minutes.

Finally, it is important to refer that with this solution there is no need to manipulate the fish, then, less stress will be induced due to manual handling.

Chapter 6

Conclusions and Future Work

6.1 Conclusions and Achievements

After the development of this project, we were able to present a noninvasive technique for zebrafish count in fish facility tanks. It was possible to develop and deliver the production of a full recording setup prototype as well as the counting software for fish number estimation, guaranteeing an error margin $\leq 15\%$ of the fish number inside the tank.

Regarding the recording prototype, we were able to produce a solution that allows the recording of videos always at the same conditions (for instance constant luminosity and distance to the camera) which is very important to guarantee the repeatability of the process.

We were able to implement computer vision techniques and mathematical tools for mirroring compensation during frames analysis, in the algorithm developed in this project, in order to output an estimate of the number of fish in a tank. At a first stage of this project, including an optical flow technique in the algorithm was considered. However, after realizing that fish are not detected when their motion is low and the fact that optical flow implicates more calculations for each frame and, consequently, more processing time, we decided to drop this idea. Hence, since the algorithm has in its base the counting of blobs resulting from background subtraction, information regarding the average area that one fish, for the different categories, represented in 7500 frames was collected. Using this information as a reference, we were able to build area intervals for different fish quantities and use that area to develop a fish overlapping compensation method. For the two different phenotypes, Nacre and AB/TU zebrafish reflect light differently since the latter are physically darker than the first which lead to the usage of different parameters for blob overlapping compensation.

Regarding performance, the algorithm demonstrated very good results particularly in fish fries categories, where the average error was always in the acceptable margin defined at the beginning of the project and the values for the variances of the error in the different video samples were very low.

In some cases, particularly in videos with adult fish, shoaling occurred and affected significantly the algorithm performance. In fact, the cases where higher errors were verified correspond to samples where shoaling occurred during almost the entire video sample or partial shoaling could be observed in a large number of frames. Due to the significant difference in size between fries and adults, overlapping, thus shoaling, tend to occur more often in adult fish population. As a matter of fact, all the results above the error margin were verified in the adults category. Nevertheless, we could conclude that in cases where shoaling does not occur, this algorithm does also demonstrate acceptable results for adult zebrafish.

After analysis of graphs such as in Figure 5.3, we could observe that after a few hundreds of frames the counting estimate was already inside the acceptable error margin which indicates that we may not need 1000 frames (40 seconds) to output an acceptable estimate.

Finally, despite the extra time that providing and estimate would take, we could also conclude that it is possible to use a Single-Board Computer (SBC) with an external USB camera (with at least HD quality) to analyse videos online, i.e, running the algorithm in real time.

6.2 Future Work

Considering an approach with only one camera as in our project and since this algorithm does already identify shoaling behaviour, frames where shoaling occurred could be skipped and not used in fish estimation. In this way we would expect to have lower error since partial shoaling and shoaling would not affect the counting. However, the disadvantage of this process would be the extra waiting time until we had enough frames without shoaling to provide an acceptable estimate.

Another relevant information that could be used to reduce the error in adults counts is previous counting records. Since at this Fish Facility zebrafish are not generally added to tanks as time passes, we could take advantage of the good results obtained in fries categories and use the counting information obtained after the first 30 days (fries) to limit the maximum number of fish that are likely to be in the tank after that time (90 days, adults). However, this would be useful to obtain an upper limit but would not, obviously, solve the shoaling issue.

This algorithm could also be implemented with two cameras: one as used in this project, and another recording on top of the tank. In this way, we could use the shoaling behaviour identification and the frame obtained from the top camera to give more accurate counting because we would have information regarding the depth of the shoal.

Another interesting and useful approach would be to use background subtraction with features detection and extraction algorithms to identify, in each tank:

- how many males or females exist
- in tanks with more than one phenotype understand how many fish exist
- study the relationship between the biomass in each tank (which would implicate fish weighing) and the average blob areas of fish in videos.

It would also be extremely useful to:

- design an implementation of the algorithm using multithreading techniques to reduce execution times in devices such as the Raspberry Pi

- measure the average time in adult zebrafish manual counting
- perform experiences in order to evaluate the manual error counting
- perform an in depth study to evaluate, quantify and compare the stress caused to fish during manual counting and automatic counting.

Bibliography

[1] R. J. Egan, C. L. Bergner, P. C. Hart, J. M. Cachat, P. R. Canavello, M. F. Elegante, S. I. Elkhayat, B. K. Bartels, A. K. Tien, D. H. Tien, S. Mohnot, E. Beeson, E. Glasgow, H. Amri, Z. Zukowska, and A. V. Kalueff. Understanding behavioral and physiological phenotypes of stress and anxiety in zebrafish. *Behavioural Brain Research*, 205(1):38–44, Dec. 2009. ISSN 01664328. doi: 10.1016/j.bbr.2009.06.022. URL `http://linkinghub.elsevier.com/retrieve/pii/S0166432809003842`. Accessed: 2016-03-06.

[2] R. Szeliski. *Computer vision: algorithms and applications*. Texts in computer science. Springer, London, 2011. ISBN 978-1-84882-935-0 978-1-84882-934-3. page 5.

[3] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. *Proceedings of the 13th Scandinavian Conference on Image Analysis*, pages 363–370, 2003.

[4] K. Stoletov and R. Klemke. Catch of the day: zebrafish as a human cancer model. *Oncogene*, 27(33):4511, July 2008. ISSN 0950-9232, 1476-5594. doi: 10.1038/onc.2008.95. URL `http://www.nature.com/doifinder/10.1038/onc.2008.95`. Accessed: 2016-04-09.

[5] Tecniplast. Tecniplast ZebraFish Tank. URL `http://www.scanbur.com/fileadmin/News/Educational_tour/ZebTEC_PPT.pdf`. Accessed: 2016-04-13.

[6] Y. Toh, T. Ng, and B. Liew. Automated fish counting using image processing. In *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, pages 1–5. IEEE, 2009.

[7] H. Khanfar, D. Charalampidis, E. Yoerger, J. Ioup, and C. Thompson. Automatic Fish Counting in Underwater Video. *66th Gulf and Caribbean Fisheries Institute*, pages 1–9, Nov. 2013.

[8] J. N. Fabic, I. E. Turla, J. A. Capacillo, L. T. David, and P. C. Naval. Fish population estimation and species classification from underwater video sequences using blob counting and shape analysis. pages 1–6. IEEE, Mar. 2013.

[9] H. A. Qader, A. R. Ramli, and S. Al-haddad. Fingerprint recognition using zernike moments, 2006.

[10] C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. B. Fisher. ”Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos”. *Proc. 3rd Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, pages 514–519, 2008.

[11] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. pages 28–31 Vol.2. IEEE, 2004. ISBN 978-0-7695-2128-2. doi: 10.1109/ICPR.2004.1333992. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1333992. Accessed: 2016-02-16.

[12] B. J. Boom, J. He, S. Palazzo, P. X. Huang, C. Beyan, H.-M. Chou, F.-P. Lin, C. Spampinato, and R. B. Fisher. A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage. *Ecological Informatics*, 23:83–97, 2014.

[13] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, May 2006. ISSN 01678655. doi: 10.1016/j.patrec.2005.11.005. URL http://linkinghub.elsevier.com/retrieve/pii/S0167865505003521. Accessed: 2016-02-16.

[14] D. Reynolds. *Encyclopedia of Biometrics*, chapter Gaussian Mixture Models, pages 659–663. Springer US, Boston, MA, 2009. ISBN 978-0-387-73003-5. doi: 10.1007/978-0-387-73003-5_196. URL http://dx.doi.org/10.1007/978-0-387-73003-5_196. Accessed: 2016-04-13.

[15] M. D. Binder, N. Hirokawa, and U. Windhorst, editors. *Encyclopedia of Neuroscience*, chapter Aperture Problem, page 159. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-29678-2. doi: 10.1007/978-3-540-29678-2_310.

[16] B. Girod, G. Greiner, and H. Niemann, editors. *Principles of 3D Image Analysis and Synthesis*. Springer US, Boston, MA, 2002. ISBN 978-1-4419-4982-0 978-1-4757-3186-6. URL http://link.springer.com/10.1007/978-1-4757-3186-6. page 35, Accessed: 2016-03-09.

[17] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, 8(2):179–187, Feb. 1962. ISSN 0018-9448. doi: 10.1109/TIT.1962.1057692. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1057692. Accessed: 2016-02-10.

[18] EGLO. 92065 | LED STRIPES-FLEX. URL http://www.eglo.com/eglo_global/Products/Main-Collections/Interior-Lighting/LED-STRIPES-FLEX/92065. Accessed: 2016-04-13.

[19] P. Foundation. Raspberry Pi 2 Model B, 2015. URL https://www.adafruit.com/pdfs/raspberrypi2modelb.pdf. Accessed: 2016-03-06.

[20] P. Foundation. Raspberry Pi Camera, 2013. URL https://www.raspberrypi.org/documentation/hardware/camera.md. Accessed: 2016-04-13.

[21] P. Foundation. Raspberry Pi 7” Touch Screen Display, 2015. URL http://www.farnell.com/datasheets/1958036.pdf. Accessed: 2016-04-09.

[22] N. Miller and R. Gerlai. From Schooling to Shoaling: Patterns of Collective Motion in Zebrafish (Danio rerio). *PLoS ONE*, 7(11):1, Nov. 2012. ISSN 1932-6203. doi: 10.1371/journal.pone.0048865. URL http://dx.plos.org/10.1371/journal.pone.0048865. Accessed: 2016-03-18.

[23] Chegg. Polar coordinates. URL http://s3.amazonaws.com/answer-board-image/1a540068-0197-4cfa-96fe-7f2daf366c5f.jpeg. Accessed: 2016-04-13.

[24] HSV Color distribution. URL http://viz.aset.psu.edu/gho/sem_notes/color_2d/gifs/hsv_model.gif. Accessed: 2016-04-13.

[25] HSV color cone, 2005. URL https://en.wikipedia.org/wiki/File%3aHSV_cone.jpg. Accessed: 2016-04-13.

[26] M. S, J. Monteiro, M. Vito, D. Weintraub, J. Almeida, and A. Certal. Toward an Integrated Zebrafish Health Management Program Supporting Cancer and Neuroscience Research, 2016.

[27] A. B. Makar, K. E. McMartin, M. Palese, and T. R. Tephly. Formate assay in body fluids: application in methanol poisoning. *Biochemical medicine*, 13(2):117–126, June 1975. ISSN 0006-2944. PMID: 1.

[28] K. J. Clark, N. J. Boczek, and S. C. Ekker. Stressing zebrafish for behavioral genetics. *Reviews in the Neurosciences*, 22(1), Jan. 2011. ISSN 2191-0200, 0334-1763. doi: 10.1515/rns.2011.007. URL http://www.degruyter.com/view/j/revneuro.2011.22.issue-1/rns.2011.007/rns.2011.007.xml. Accessed: 2016-04-08.

Appendix A

Fish Counting - Extra Tables and Graphs

Fish Number	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	Average Error	Average Error [%]
5	1	0	1	1	0	1	1	1	2	0	1	1	1	0	0	0	0	1	1	1	0,7	14
7	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1	2	1	0	0,85	12
10	1	1	1	2	1	1	2	2	1	2	2	1	1	2	2	1	1	2	2	2	1,5	15
12	0	1	1	1	0	1	1	0	1	0	2	1	0	2	0	0	1	1	1	1	0,75	6
15	1	2	1	2	1	1	1	0	1	1	2	1	1	1	0	2	1	2	2	1	1,2	8
17	1	2	1	1	1	1	0	1	1	1	0	2	2	1	0	0	1	1	1	1	0,95	6
20	2	2	2	2	2	2	3	2	2	2	1	2	2	2	3	3	3	2	2	2	2,15	11
22	1	0	0	1	1	1	2	1	2	1	1	0	0	0	0	0	1	0	0	1	0,65	3
25	1	0	2	1	1	0	1	1	2	1	1	2	1	2	0	1	0	0	1	2	1	4
27	2	2	1	1	1	2	2	1	2	2	1	2	1	1	1	1	1	2	1	1	1,4	5
30	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0,25	1
32	3	3	2	2	3	3	4	2	1	4	3	4	3	2	0	2	2	2	3	2	2,5	8
35	3	0	2	2	2	2	1	3	2	0	0	0	1	0	0	3	2	1	2	3	1,45	4

Table A.1: Absolute error in each of the 20 video samples and average counting error for every school of fish in Nacre fries category, where red represents average error above 15% of the real fish value and green the opposite.

Fish Number	Variance	Standard Deviation
5	0,01	0,09
7	0,04	0,19
10	0,03	0,16
12	0,03	0,18
15	0,004	0,06
17	0,0003	0,02
20	0,002	0,05
22	0,01	0,11
25	0,05	0,22
27	0,03	0,16
30	0,01	0,08
32	0,03	0,16
35	0,24	0,49

Table A.2: Variance and standard deviation in 20 video samples for every school of fish in Nacre fries category.

Fish Number	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	Average Error	Average Error[%]
5	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0,2	4
7	1	0	1	1	1	1	1	2	1	1	0	1	1	1	0	1	2	0	1	1	0,9	13
10	0	1	1	1	0	1	0	1	1	1	1	1	1	0	1	0	1	0	0	1	0,65	7
12	2	2	1	1	1	2	1	1	2	1	1	1	0	1	1	2	1	1	2	1	1,25	10
15	0	0	2	1	1	1	1	0	1	0	0	1	0	0	0	1	1	0	0	1	0,55	4
17	2	2	2	2	2	2	1	2	2	3	1	1	2	1	1	0	1	1	0	2	1,5	9
20	0	2	3	3	1	3	1	3	3	0	0	1	1	0	0	0	1	2	2	1	1,35	7
22	3	3	2	2	3	3	2	1	2	1	4	2	1	1	1	2	2	1	2	2	2	9
25	1	0	0	2	1	0	0	1	2	1	1	1	1	1	2	0	2	0	0	2	0,9	4
27	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	0	0,75	3

Table A.3: Absolute error in each of the 20 video samples and average counting error for every school of fish in AB/TU fries category, where red represents average error above 15% of the real fish value and green the opposite.

Fish Number	**Variance**	**Standard Deviation**
5	0,03	0,18
7	0,00	0,03
10	0,03	0,17
12	0,03	0,18
15	0,03	0,16
17	0,03	0,16
20	0,10	0,31
22	0,05	0,22
25	0,06	0,25
27	0,03	0,18

Table A.4: Variance and standard deviation in 20 video samples for every school of fish in AB/TU fries category.

Fish Number	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	Average Error	Average Error[%]
5	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1,05	21
7	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0,9	13
10	1	1	1	0	3	2	2	0	2	3	1	3	4	2	1	2	1	3	1	3	1,8	18
12	0	3	2	0	3	0	0	2	1	4	1	1	1	2	1	0	0	4	0	4	1,45	12
15	4	7	4	4	4	5	2	7	2	1	5	4	2	5	5	2	7	4	7	6	4,35	29
17	1	2	3	0	1	3	2	0	0	0	1	4	1	0	0	0	0	1	1	1	1,05	6
20	2	1	1	3	2	3	3	0	3	0	6	1	4	1	3	0	1	1	2	0	1,85	9
22	9	2	6	7	6	6	1	1	9	8	2	2	1	2	8	3	6	0	8	2	4,45	20
25	2	0	0	4	2	3	0	2	5	4	1	3	5	4	2	1	1	0	1	3	2,15	9
27	3	2	4	4	4	3	5	3	4	0	2	1	4	2	3	1	3	3	4	5	3	11
30	2	1	0	2	1	1	3	3	1	2	2	3	2	2	2	4	2	1	4	1	1,95	7
32	2	1	0	1	0	0	1	3	2	0	0	1	0	2	0	1	1	3	1	1	1	3
35	2	0	2	0	4	2	3	1	2	1	4	1	2	1	1	0	1	1	2	1	1,55	4

Table A.5: Absolute error in each of the 20 video samples and average counting error for every school of fish in Nacre adults category, where red represents average error above 15% of the real fish value and green the opposite.

Fish Number	Variance	Standard Deviation
5	0,00	0,02
7	0,00	0,03
10	0,10	0,32
12	0,43	0,66
15	0,14	0,38
17	0,00	0,02
20	0,17	0,42
22	1,34	1,16
25	0,04	0,19
27	0,20	0,45
30	0,05	0,21
32	0,05	0,22
35	0,03	0,16

Table A.6: Variance and standard deviation in 20 video samples for every school of fish in Nacre adults category.

Fish Number	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	Average Error	Average Error[%]
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	1	0	1	0	1	0	1	0	0	0	1	1	0	0	1	0	0,4	6
10	0	1	0	1	1	0	0	1	0	0	1	0	0	0	0	1	1	1	1	1	0,5	5
12	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	0	0	1	0	0	0,3	3
15	1	2	2	1	2	1	2	1	1	2	2	2	1	1	1	2	2	1	1	0	1,4	9
17	2	3	3	2	1	2	2	2	2	2	2	2	2	2	1	3	2	1	1	1	1,9	11
20	4	3	2	3	2	3	2	2	2	3	1	4	4	5	2	2	2	2	4	4	2,8	14
22	0	2	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0	1	0,75	3
25	0	4	2	0	2	0	1	2	0	2	1	0	2	2	1	1	0	1	1	2	1,2	5
27	4	8	6	8	9	8	9	6	9	5	9	9	5	1	8	1	8	9	4	3	6,45	24
30	2	5	3	3	5	4	8	5	4	4	7	1	5	3	7	4	2	9	4	7	4,6	15
32	10	3	2	2	1	3	2	11	9	8	2	3	3	3	2	2	3	3	3	2	3,85	12
35	10	12	11	12	7	8	10	12	10	8	10	10	10	10	12	11	11	11	9	11	10,25	29

Table A.7: Absolute error in each of the 20 video samples and average counting error for every school of fish in AB/TU adults category, where red represents average error above 15% of the real fish value and green the opposite.

Fish Number	Variance	Standard Deviation
5	0,00	0,00
7	0,03	0,16
10	0,03	0,16
12	0,01	0,09
15	0,11	0,33
17	0,04	0,20
20	0,14	0,38
22	0,03	0,18
25	0,10	0,32
27	0,90	0,95
30	0,63	0,79
32	2,06	1,44
35	0,03	0,18

Table A.8: Variance and standard deviation in 20 video samples for every school of fish in AB/TU adults category.

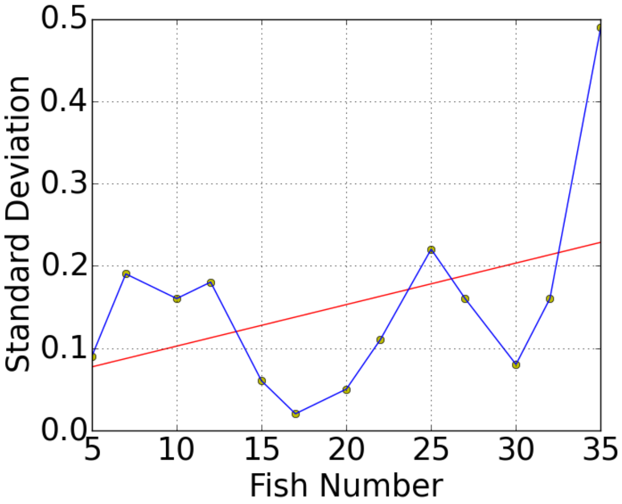


Figure A.1: Plotted standard deviation for Table A.2.

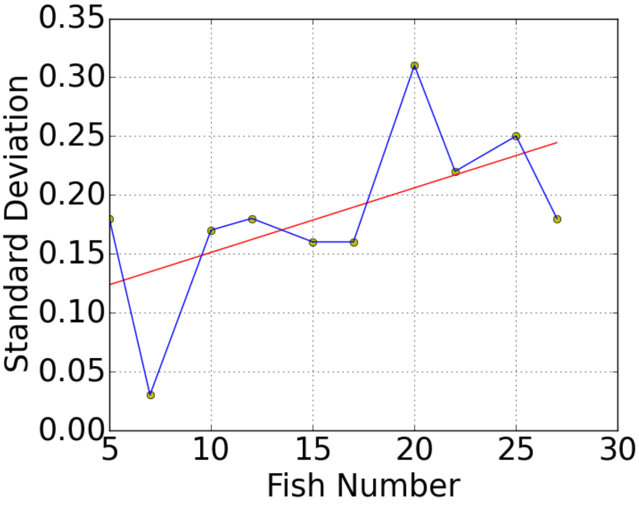


Figure A.2: Plotted standard deviation for Table A.4.

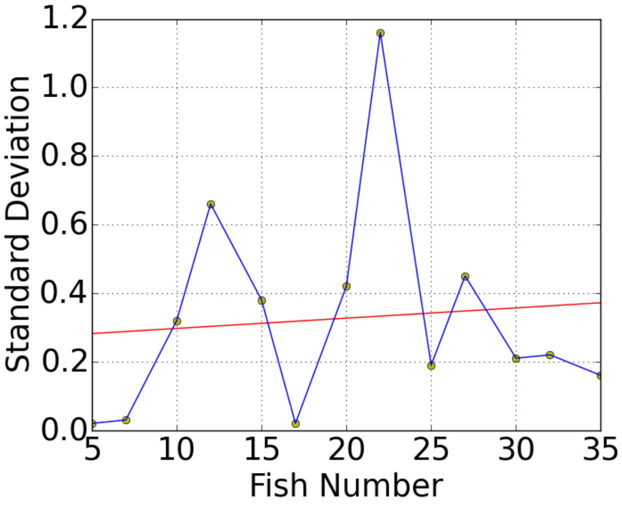


Figure A.3: Plotted standard deviation for Table A.6.

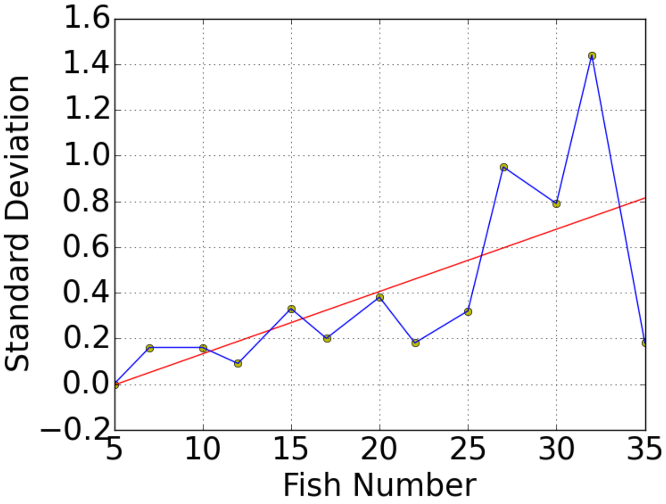


Figure A.4: Plotted standard deviation for Table A.8.