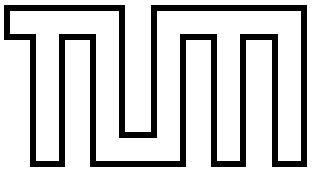


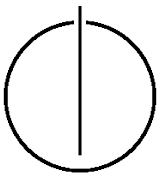
FAKULTÄT FÜR INFORMATIK

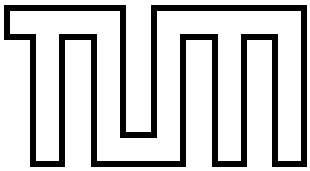
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Mater's Thesis in Biomedical Computing

Deformable object detection in underwater imaging

Andrés Sánchez





FAKULTÄT FÜR INFORMATIK

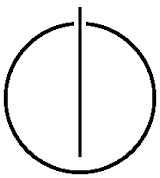
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Mater's Thesis in Biomedical Computing

Deformable object detection in underwater imaging

Deformierbare Objekterkennung in Unterwasser-Bilder

Author:	Andrés Sánchez
Examiner:	Prof. Dr. Nassir Navab
Supervisor:	Prof. Dr. Slobodan Ilic
Advisor:	M.Sc. David J. Tan
Date:	November 27, 2013



I hereby declare that this thesis is enterely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

München, den 7. Juli 2014 Andrés Sánchez

Acknowledgments

I have a great deal of people to thank for the presented work, but the most important one is my wife, who left everything behind in our home country to be with me in this adventure, Without her company, support and affection, I could not fulfill my dream of studying in Germany. Also, I would like to thank my family for their outstanding support for the entire duration of my course.

I thank Dr. Slobodan Ilic and Prof. Dr. Nassir Navab as they play important roles in the completion of this thesis. In addition, I acknowledge the help from the Computer Vision group at the Chair for Computer Aided Medical Procedures and Augmented Reality. Furthermore, I would like to specifically mention the contribution of David Tan, for sharing his knowledge during our project in the field of deformable object detection in underwater applications.

Also, I would like to thank to three compatriot, who make my stay in the TU München possible, Dr. Víctor Castañeda for help me during my application to the program and the beginning of this process, José Gardiazabal for offering always a hand when I need it and Eduardo Morral to offered me accommodation during the tough task of find a place to live in München.

Finally, it is a pleasure to thank who made this possible. I want to thank the Chilean National Commission for Science and Technology $(CONICYT)$, whose funding and confidence have done possible to carry out my project in Germany.

Abstract

The monitoring of fish for stock assessment in aquaculture, commercial fisheries and in the assessment of the effectiveness of biodiversity management strategies such as marine protected Areas and closed area management has been thriving since the 1980s. as does area continuously grows, it becomes important to develop a remote monitoring system to estimate the biomass of the large number of fishes bred in cages, since around 80% of all sales of farmed fish are arranged pre-harves, that mean, the profit on the sale directly depends on correct estimations of weight, size distribution and total biomass. Therefore automated and relatively affordable tools for biomass estimation have to be developed.

Here, we will rely on complex stereo camera system, compose of time of flight range camera and CCD grayscale camera, that film fishes in the cage for certain period of time. in order to estimate the biomass, the volume of the fish has to be estimated. this can be achieved by first detecting and segmenting the fish in every grayscale image of the incoming video stream and then translate this found fish contour to the range image obtaining a estimation of the volume. To find the algorithm that is in line with our problem, we need to understand the challenge in detecting fishes. they include the motion of the fish which makes the object of interest deformable, the location of the fish respect to the camera and occlusions caused by having multiple fishes in every available frame.

In this project, we concentrate on the first step that is detection of the fish that undergo deformation in grayscales images. we propose an approach inspired by two works from, the first, Yang and Ramanan which describe a method for articulated human detection and human pose estimation in static images based on a representation of deformable part models. The main idea of their approach is to use a mixture of small, non-oriented parts, which describe a general, flexible mixture model that jointly captures spatial relations between parts locations and co-occurrences relations between part mixtures, augmenting standard pictorial structure models that encode just spatial relations. The second approach is propose in Hinterstoisser et al. [2012], where they present a method for real-time 3D object instance detection that does not require a time consuming training stage, and can handle untextured objects. At its core, the approach presented is a novel image representation for template matching designed to be robust to small image transformations, This robustness is based on spread image gradient orientations and allows to test only a small subset of all possible pixel locations when parsing a image.

We evaluate the proposes method by computing difference between the labeled dataset with the predicted result, in addition, we cluster the results from different camera locations and found that when the sagittal plane is parallel to the image plane, the tracking algorithm provide the best result.

Therefore, In this thesis, we accomplished the goals of creating a annotated datasets that comprise of learning keypoints and fish contours from a set of 2D grayscale images. We also implement a combine solution from a part based detection model and a template matching detection approach, which is capable of predict keypoints and fish contours in an unlabeled 2d grayscale image and verifying the validity of the prediction.

Contents

Outline of the Thesis

Part I: Overview

CHAPTER 1: INTRODUCTION

This chapter presents the motivation and the problem that we are trying to solve. It gives the context of the whole thesis.

CHAPTER 2: RELATED WORKS

This chapter is an ensemble of publications that are related to the proposed approach.

Part II: Methods and Implementation

CHAPTER 3: METHODS

This chapter present the mathematical computations, derivations and theorems as well as the formation of algorithms that are used in the thesis.

CHAPTER 4: IMPLEMENTATION

This chapter shows the implementation details of the defined Methods.

Part III: Results and Conclusion

CHAPTER 5: RESULTS AND DISCUSSION

This chapter test the algorithm and discuss the results.

CHAPTER 6: CONCLUSION

This chapter presents the author's final thoughts that includes the conclusion and future work.

Part I.

Overview

1. Introduction

1.1. Motivation

As we progress from livelihood fisheries to aquaculture industries, the global production and demand of fishes has drastically increased over several decades. According to Asche and Bjorndal [2011], the production increased from 16 M in the 1970s to 142 M in 2008. In these statistical figures, the amount if wild fishes has reached a threshold since 1980s while the farmed fishes picked up the difference in amount. For instance, LARSEN and ASCHE mention in [2011] that Norway alone increased their production of Salmons from a few thousand in the 1980s to approximately 1.4 M in 2009 which constitutes around 51% of the global supply. This makes then the largest supplier of Salmons in the world [Asche and Bjorndal, 2011; LARSEN and ASCHE, 2011; Liu et al., 2011].

Other than favourable geographical and environmental features that made Norway viable for this industry technological advancement also played an important role in the economical cycle between demand and supply. As production increase, they reduced cost and as a consequence, increased the demand Asche and Bjorndal [2011]. Therefore, this cycle supported the growth of the industry over the years.

Since around, 80% of all sales of farmed fish are arranged pre-harvest, the profit on the sale directly depends on the correct estimations of weight, size distribution and total biomass. Therefore, our project deals with remote monitoring of fishes size and weight distribution in aquaculture environments. Considering a large amount of fish, it becomes essential to develop an automated biomass estimator to constantly monitor the changes or growth of fishes. This system involves cameras that would detect the fish in a video sequence and compute the biomass distribution over a specified period of time.

1.2. Problem Statement

This work is part of the project call fishscan, where the main goal is design a system for remote monitoring of fishes size and weight distribution in aquaculture environments. during this project was develop a camera rig system consist in a underwater housing with a time of flight camera with LED light source and a 2D CCD grayscale camera as is shown in the fig. 1.1.

As the main goal of the project is compute the biomass of the fish by computing the volume of it, taken the concept of mass density from the physics, using the relation between biomass and volume. Then, this problem of biomass estimation can be formulated as a problem of volume estimation of the fish. to achieve this objective, the first step is the fish detection in an 2D grayscale image. followed by a back-projection into the TOF image where the is possible to fit a 3D model to the detected fish. it is important to mention that the approach assume in this work is due of highly noisy image acquire by the range imag-

Figure 1.1.: Rig Camera System - TOF + CCD cameras

ing camera, which was adapted to work in a underwater environment, but as you can see in the 3D image shown in Fig. 1.2

although, the detection using the 2D intensity image alone cannot compute the volume of the fish because it is not depth invariant and the size is up-to-scale; This work will may use of the 2D intensity image as a First step, detecting the fish contour. The pipeline depicted in Fig. 1.3 consist of three major steps that are: fish detection, contour extraction, volume-biomass estimation. In this project, we concentrate on the first two steps that is fish detection and contour extraction.

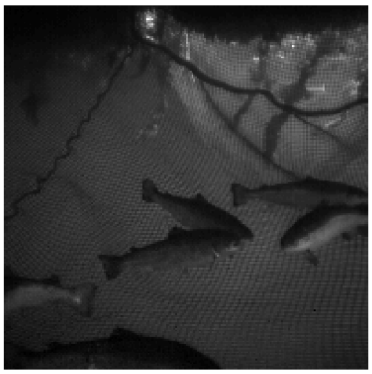
At this point we need to find a algorithm for fish detection and contour extraction that addresses the three major challenges of our problem, These are:

1. *Deformations*, The algorithm must handle different motions of the which suggests that we are dealing with a deformable (articulated) object.

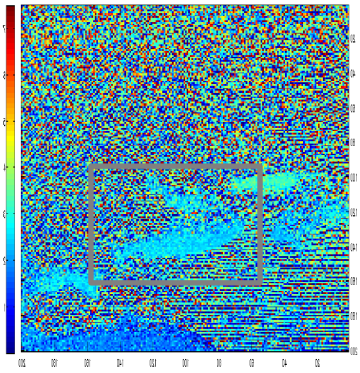
2. *Different Viewpoints*, As the fish move around its environment, the algorithm must be able to detect the fish from different perspectives.

3. *Occlusions*, The algorithm must also be able to handle occlusion, e.g. self occlusions, occlusion from another fishes and occlusion from object in the environment.

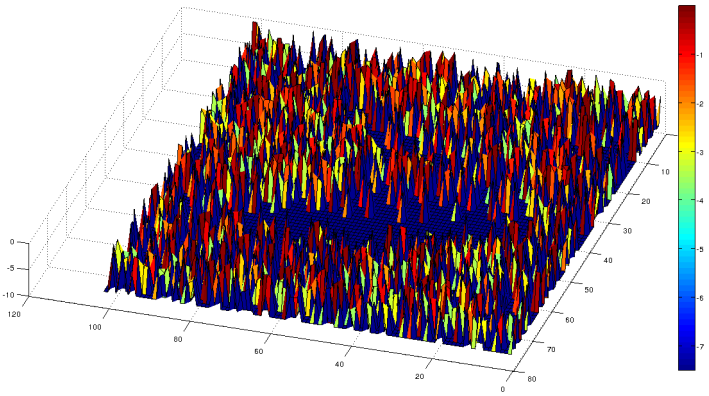
Based on the three challenges and the use of 2D intensity values obtained from the 2D CCD camera, we propose an approach inspired by two works from, the first, Yang and Ramanan which describe a method for articulated human detection and human pose estimation in static images based on a representation of deformable part models. The main idea of their approach is to use a mixture of small, non-oriented parts, which describe a general, flexible mixture model that jointly captures spatial relations between parts locations and co-occurrences relations between part mixtures, augmenting standard pictorial structure models that encode just spatial relations. The second approach is propose in Hinterstoisser et al. [2012], where they present a method for real-time 3D object instance detection that does not require a time consuming training stage, and can handle untextured



(a) Intensity Image TOF camera



(b) depth Image



(c) 3D represemtation depth image

Figure 1.2.: TOF camera images.

objects. At its core, the approach presented is a novel image representation for template matching designed to be robust to small image transformations, This robustness is based on spread image gradient orientations and allows to test only a small subset of all possible pixel locations when parsing a image. In our approach, that can be consider in the groups of machine learning method, the learning process tries to understand the relation between the input X and output Y; such that when the input X is given, it can predict the outcome Y. In our case, the input X are 2D grayscale intensity image containing fishes under different deformations and seen from different perspectives, while the expected output Y are the positions of the desired keypoints and corresponding contours for the detected fishes, However, in the learning stage, we require a large amount of data that shows this relations. We need a great amount of labeled 2D intensity image where the locations of the keypoints and contours are given. unfortunately, the current motion tracking systems from human pose estimation are not a valid solution to find ground truth data for fish because the difference in behaviour as well as the difference in environment. Therefore, we address this

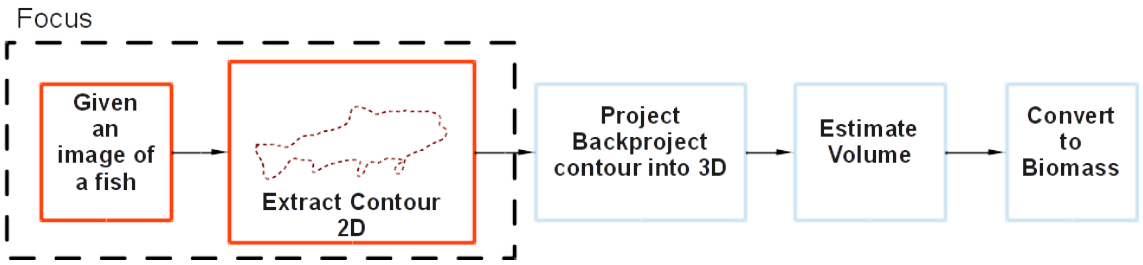


Figure 1.3.: The framework that describe the overview of the problem

problem by hand labeling real 2D deformed fishes images of a group of in a real scenario, where the fish are observed from different perspectives.

We evaluate the proposed method by the broadly-adopted evaluation protocol based on the probability of a correct pose (PCP), which measure the percentage of correctly localized body parts, a candidate body part in labeled as correct if its segment endpoints lie within 50% of the length of the ground-truth annotated endpoints.

Therefore, this thesis accomplished the following: create a labeled real fish images datasets that comprise of intensity images acquire by 2D CCD camera, learn esqueleton and countours from labeled datasets, predict keypoints and contours an unlabeled 2D fish images and verify the validality of the predictions that will be discussed in Chapter 3. A overview of the related works is in Chapter 2 while the implementations details are presented in Chapter 4. The result is discussed in Chapter 5 and finally, we conclude in Chapter 6.

2. Related Work

2.1. Related Work

This thesis focuses on detecting fishes; therefore, this chapter concentrates on works that deal with object detection, specifically in underwater application. In Shortis et al. [2013] reviewed the state of arts on the field of identification and measurement of fish, using underwater stereo-video sequences. The most interesting approaches describes in this paper, are related with automated measurement of fish in video sequences as describe by Tillett et al. [2000] in what is one of the first published reports on successful, operational, automated measurement system. The technique is based on 3D Point Distribution (PDM), which are composed of landmark locations on the outline of the fish, in this case Atlantic Salmon held in a small aquaculture tank. The PDM specific to the species is developed from a small sample of fish defined by manual measurement of stereo-images, leading to a mean shape and an estimate of the variation based on principal component analysis. The PDM is independent of the scale and orientation, but is limited only to the silhouette of the fish and does not model the full body shape. This work, also analyse the methodologies for detection of fishes which comprises two steps: Identification and subsequent delineation of the fish outline. Most of the existing work on fish detection employ either the differences between successive images Spampinato and Chen-Burger [2008]; Tillett et al. [2000] or histogram-thresholds to segment a varying number of candidate regions in the frames. Active contours(also called snake) are especially useful for delineating objects like fish bodies that are difficult to model with rigid geometric primitives. Moreover, active contours can be independent from edge gradients with flexibility in initialisation [Chan and Vese, 2001]. The area-based active contour model Chan and Vese [2001] is based on the on techniques of curve evolution and level sets. While parametric active contours cannot handle automatic change of topology, level sets allow for splitting and merging in a natural way and are thus more suited for detection of an unknown number of fish in a video image. In Shortis et al. [2013], they also tackle the different technique apply in measurement. Underwater stereo systems are widely used to capture video of swimming fish for subsequent measurement. the simplest form of measurement is the fish snout to tail length which can be calculated if these two points can be identified in the stereo pair of images. in our days this is done manually in most cases, with a favourable orientation of the fish to the cameras and a multiple measurements within in the sequences of frames do improve the precision of the measurements. One important point to be noted is that fish are deformable and the euclidean distance from snout to tail changes as the fish swims. Template matching is one of the primitive methods that can be employed to accurately locate fish snout and tail in video frames. First, individual templates (usually rectangular image regions) centered on the snout and tail mid-points are extracted from sample videos. Then an efficient template matching strategy is employed to locate these templates in target videos. A certain degree of robustness against illumination changes can be achieved

by using correlation between template and image regions of interest, instead of taking their absolute differences [Mahmood and Khan, 2012]. However, template based methods fail in the presence of perspective or affine transformations, requiring either use of multiple templates that capture appearance variations from different viewing angles, or using more sophisticated matching techniques that are invariant to affine or perspective transformations. These enhancements also significantly increase computational complexity of the template matching step. A better way of locating snout and tail is to use Haar-like features in a boosted classifier setup [Viola and Jones, 2001] that has shown high object detection accuracy, besides being able to operate in real-time. The method is in wide use for face detection. To train the classifier, manually cropped images of the target object (snout or tail) are used so that the classifier can learn which features (among a set of possibly thousands of features) can locate the target with high accuracy. These features, once learned, are then used to construct the object classifier that can locate the presence of the object in cluttered scenes. Due to their high detection speed and ability to perform a scale-space search, Haar classifiers are a promising candidate for locating snout and tail of fish in underwater images. The results of independent detection of the snout and tail using Haar detectors can be further improved using relationships between the detected snouts and tails, for instance by constraining the search for tail detection based on the results of snout detection and vice versa.

Furthermore, since we are detecting fishes, it is safe to limit our scope to articulated object detection. This lead to the idea that our problem is similar to human pose estimation. and specifically two works from, the first, Yang and Ramanan which describe a method for articulated human detection and human pose estimation in static images based on a representation of deformable part models. The main idea of their approach is to use a mixture of small, non-oriented parts, which describe a general, flexible mixture model that jointly captures spatial relations between parts locations and co-occurrences relations between part mixtures, augmenting standard pictorial structure models that encode just spatial relations. The second approach is propose in Hinterstoisser et al. [2012], where they present a method for real-time 3D object instance detection that does not require a time consuming training stage, and can handle untextured objects. At its core, the approach presented is a novel image representation for template matching designed to be robust to small image transformations, This robustness is based on spread image gradient orientations and allows to test only a small subset of all possible pixel locations when parsing a image.

Part II.

Methods and Implementation

3. Methods

The aim of this chapter is to define a basic workflow for the new detection approach and provide the reader with relevant literature review. The workflow will be divided in to different functional processes. The requirements (Input) and the outcome (Output) of each process will be defined. Based on these requirements relevant theorical concepts will be discussed. The processes directly related to the problem statement will be discussed in detail, and a suitable approach will be suggested. The reader should note that this chapter will provide only the overview of the theorical concepts, implementation specific details will be covered in chapter 4 of the thesis.

To decide the best way to approach this problem, we need first know our target for the detection algorithm. In this case specifically, as we already mentioned, are fishes within aquaculture enclosures. To understand in a better form our target, we begin explaining the fish swimming mechanics, this will bring the necessary knowledge to define the strategy and possible constraint to our propose algorithm.

3.0.1. Fish swimming mechanics

A basic consideration for the design of algorithm to detect deformable object like fish are their shape and pattern of movement. Studies have identified several types of locomotion that fish use to generate thrust Hawkes et al. [2008]; Colgate and Lynch [2004]. Most fish generate thrust by bending their bodies into a backward-moving propulsive wave that extends to the caudal fin, a type of swimming classified under body and/or caudal fin (BCF) locomotion. The propulsive wave traverses the fish body in a direction opposite to the overall movement and at a speed greater than the overall swimming speed. There are four undulatory BCF locomation modes indetified by their amplitude envelope of the propulsive wave: *anguilliform, subcarangiform, carangiform and thunniform*. Despite these labels placed by biologists, two dimensions analyses of the fish locomotion have shown that even fishes of very different body types show extremely similar patterns of body movement when viewed in a horizontal section during steady undulatory locomotion. Nevertheless, *subcarangiform* swimming mode is the basis of the undulatory motion that present the species which are farming in aquaculture project, like the Salmon and Trout.

from the Figure 3.0.1, what interest to us is the third class, the *Carangiform* and *subcaringform* locomotion model is describe as a movement where only the posterior half of the body flexes with the passage of the contraction waves.

as shown in Figure 3.0.1, most part of the deformation present in the fish happen in the tail, and trough the backbone fish axis. After review the physiological fish model, it is possible to infer that our algorithm should model the fish, basically as a two part model, considering as a head-tail model. The next section will introduce our propose approach.

[1]Source: http://esi.stanford.edu/exercise/exercise4.html

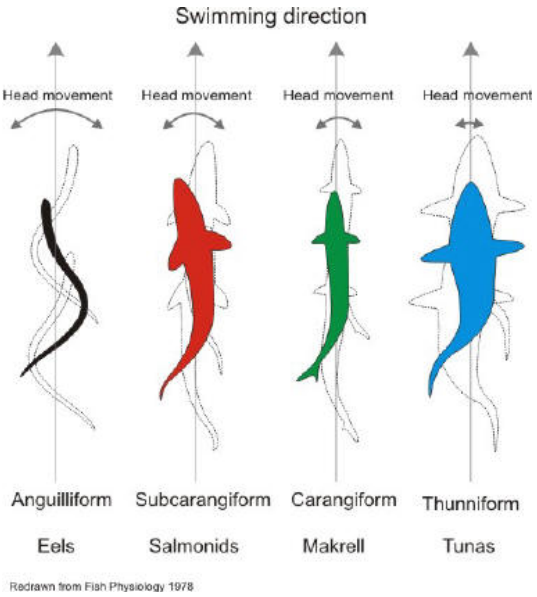


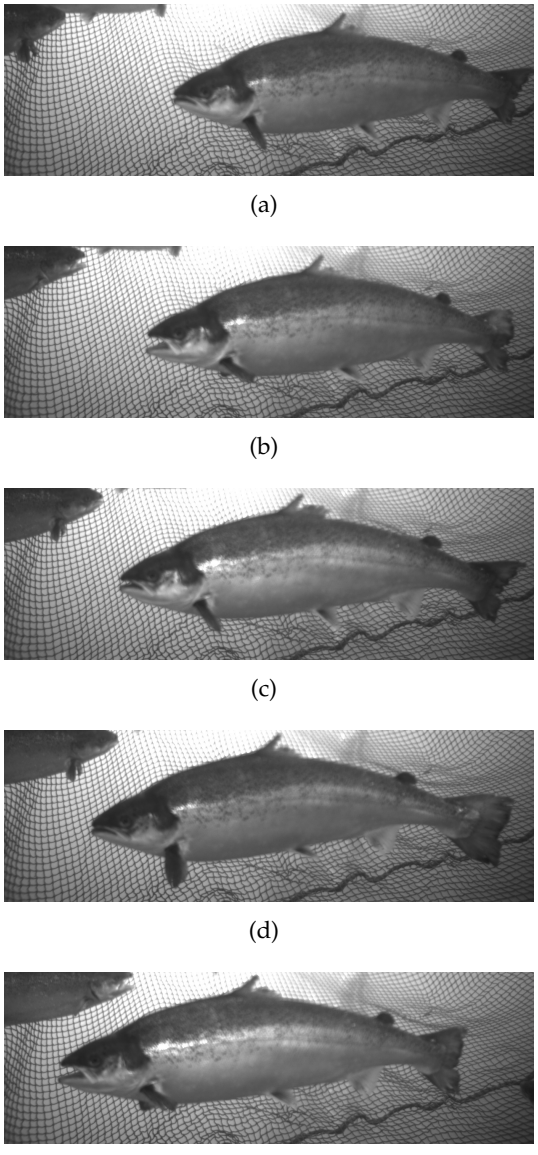
Figure 3.1.: This illustrate the classifications for fish locomotion, In our project the most interested is the *subcaringform*.[1]

3.1. Notation and Symbols

This section introduces the common mathematical notations and symbols used in this chapter. Normal formatting such as a and A are used to indicate integers or real numbers while letters in scripts such $\mathcal{A}$ are reserved for sets. In addition, we use the symbols $\mathbb{R}$ for real numbers. For 3D vectors, we use the uppercase bold letters $\mathbf{A}$ while for 2D vectors, we use the lowercase bold letters $\mathbf{a}$. If the vector is homogeneous, it will be explicitly define; otherwise, the vector is assumed to be inhomogeneous. A vector from point A to B is presented as $\overrightarrow{AB}$. Furthermore, matrices use monospace font such as $\mathtt{A}$. If the dimension are specified such as $\mathtt{A}_{mxn}$, m would indicate the number of rows while n indicates the numbers of columns. Regardings accents, a hat on a vector such as $\hat{a}$ or $\hat{A}$ indicates the normalized unit vector which means that $\hat{a} = \frac{a}{\|a\|}$ or $\hat{A} = \frac{A}{\|A\|}$. A tilde on a 3D vector indicates that the last coordinate is removed; thus, the vector $A = (x, y, z)^T$ have $\hat{A} = (x, y)^T$ which is the projection of A in the xy-axis, while the vector with homogeneous coordinate $B = (x, y, z, 1)^T$ have $\hat{B} = (x, y, z)^T$ which is the inhomogeneous coordinate of B. Lastly, a dot on top of variable indicates the converged value of the variable after an algorithm is performed. For instances, after using mean shift on a_i, it converges to a value $\dot{a}_i$

3.2. Theorical Background and Propose Workflow

Figure. 3.3 shows the basic workflow of the detection process. The figure shows that once image is captured, it is supplied to the Part Based model detection process to extract 2D bounding boxes from the define part based fish model. Also from this process can be extracted the skeleton of the fish, After 2D mask are generated, template detection matching



(a)

(b)

(c)

(d)

(e)

Figure 3.2.: This illustrates a Salmon swimming in a aquaculture enclosure, here is possible to see that the tail present most part of the deformation in the fish body

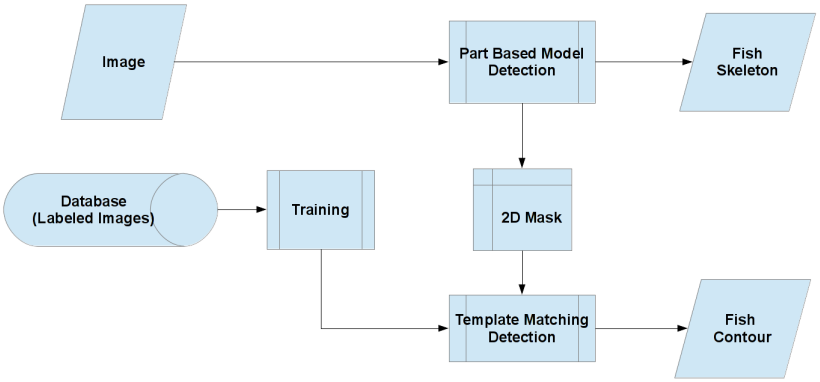


Figure 3.3.: A simplified workflow of the Detection process

process will take the 2D masked image and in those defined areas will match the learned templates using the linemod algorithm propose in Hinterstoisser et al. [2012]. It is assumed that two detection process has been trained with the proper labeled fish images in a offline procedure. A short introduction to the terms, processes and their functionalities is given below.

- *Image*, this term refers to the two dimensional photographic image of the object captured by the camera. It is given as a input for the detection workflow. The reader should note that the image provided to the workflow come from the underwater rig system develop as part of the project. which suggests that we are dealing with a deformable (articulated) object.

- *Database*, This term represents all the preprocessed information that is readily available to the detection workflow. This information includes details regarding labeled data for training, trained model for Part Based detection process, and trained model for template matching process.

- *2D Mask*, this term refer to the spatial filter, which in our problem keep the spatial information result from the part based algorithm.

- *Part Based Model Detection*, this term refers to a broad class of detection algorithms used on images, in which various parts of the image are used separately in order to determine if and where an object of interest exists. Amongst these methods a very popular one is the constellation model which refers to those schemes which seek to detect a small number of features and their relative positions to then determine whether or not the object of interest is present.

- *Template Matching Detection*, This term refers a broad class of algorithm for finding small parts of an image which match a template image. this approach can be divide in Feature-based and Template-based.

- *Training - Supervised Learning*, This term refers to the machine learning task of inferring a function from labeled training data. The training data consist of a set of samples, where each sample is a pair of a input object and a desired output value.
- *Fish skeleton*, Correspond to simplified representation of the fish, as a set of stick joining specific features in the body.
- *Fish Contour*, correspond to a curve along the fish silhouette

3.2.1. Part Based model Detection

In this section will be discuss the method used to achieve the first task define in 3.2, analysing our target object, the fish, which can be model like a deformable object with one main deformation axis as show in the Figure 3.0.1, at the beginning of this chapter, the one define by the backbone, Most fish move by alternately contracting paired sets of muscles on either side of the backbone. These contractions form S-shaped curves that move down the body.

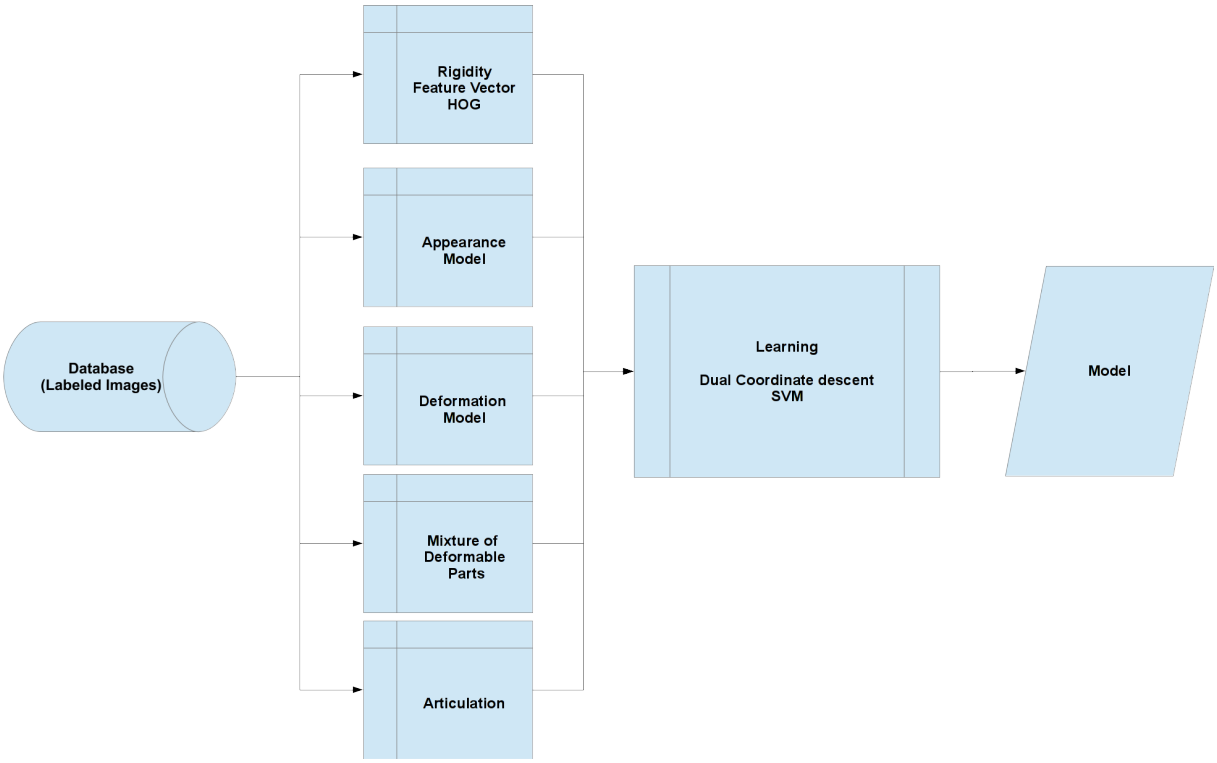


Figure 3.4.: Simplified representation of Part based model detection - Mixtures of parts

The pictorial structure framework arise as a feasible approach, which decomposes the appearance of the objects into local part templates, together with geometric constraints on pairs of parts, often visualized as a spring. when parts are parametrized by pixel location and orientation, the resulting structure can model articulation. This has been the dominant approach for human pose estimation. For our problem comparing with Full-body pose

estimation where many degrees of freedom has to be estimated, The fish is a simplification due to the movement constraints, and also the absence of big limbs which are replace by small fins, those fins not vary greatly in appearance in compare with human limbs. in the broad class of part based detection algorithm, the one propose by **?** arise as the state of the arts, and is the one selected to be applied in our problem.

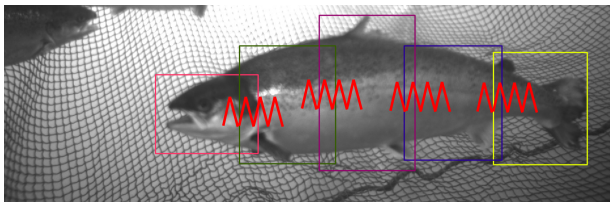


Figure 3.5.: The flexible mixture-of-parts model approximate small warps by translating patches connected with spring. Hence, this model captures the dependence of local part appearance on geometry.

Model

Let us write I for a image, $l_i = (x, y)$ for the pixel location of part i and t_i for the mixture component of part i. We write $i \in \{1 \dots K\}$, $l_i \in \{1 \dots L\}$ and $t_i \in \{1 \dots T\}$. we call t_i the "type" of part i. Our motivating examples of types include orientations of parts (e.g., a vertical versus horizontally oriented fin), but types may span out-of-plane rotations (front-view head versus side-view head) or even semantic classes (an open versus side-view head) or even semantic classes (an open versus closed hand). For notational convenience, we define lack of subscript to indicate a set spanned by that subscript (e,g,. $t = \{t1 \dots t_K\}$. For simplicity, we define our model at a fixed scale; at test time we detect fish of different sizes by searching over an image pyramid.

Co-occurrence model: To score of a configuration of parts, we first define a compatibility function for part types that factors into a sum of local and pairwise scores:

$$S(t) = \sum_{i \in V} b_i^{t_i} + \sum_{ij \in E} b_{ij}^{t_i, t_j} \tag{3.1}$$

The parameter $b_i^{t_i}$ favors particular type assignments for part i, while the pairwise parameter $b_i j^{t_i, t_j}$ favors particular co-occurrences of part types. For example, if part types correspond to orientations and part i and j are on the same rigid limb, then $b_i j^{t_i, t_j}$ would favor consistent orientation assignments. Specifically, $b_i j^{t_i, t_j}$ should be a large positive number for consistent orientations t_i and t_j, and a large negative number for inconsistent orientations t_i and t_j.

Rigidity: we write $G = (V, E)$ for a (tree-structured) $K - node$ relational graph whose edges specify which pairs of parts are constrained to have consistent relations. Such a graph can still encode relations between distant parts through transitivity. For example, our model can force a collection of parts to share the same orientation, so long as the parts to share the same orientation, so long as the parts form a connected $subtree$ of $G = (V, E)$. We use this property to model multiple parts on the torso. Since co-occurrence parameters

are learned, our model learns which collections of parts should be rigid. We can now write the full score associated with a configuration of part types and positions:

$$S(t,l,t) = S(t) + \sum_{i \in V} \omega_i^{t_i} \cdot \phi(I, l_i) + \sum_{ij \in E} \omega_{ij}^{t_i,t_j} \cdot \psi(l_i - l_j) \tag{3.2}$$

where $\phi(I, l_i)$ is a feature vector(e.g, HOG descriptor Dalal and Triggs [2005]) extracted from pixel location l_i in image I. we write $\psi(l_i - l_j) = [dx\ dx^2\ dy\ dy^2]^T$, where $dx = x_i - x_j$ and $dy = y_i - y_j$, the relative location is defined with respect to the pixel grid and not the orientation of part i (as in classic articulated pictorial structures).

Appearance model: The first sum in 3.2 is an appearance model that computes the local score of placing a template $\omega_i^{t_i}$ for part i, tuned for type t_i, at location l_i.

Deformation model: The Second term can be interpreted as a "switching" spring model that controls the relative placement of part i and part j by switching between a collection of springs. Each spring is tailored for a particular pair of types (t_i, t_j), and is parametrized by its rest location and rigidity, which are encoded by $\omega_{ij}^{t_i,t_j}$.Our switching spring model encodes the dependence of local appearance on geometry, since different pairs of local mixtures are constrained to use different springs. Together with the co-occurrence term, it specifies an image-independent "prior" over part locations and types.

Mixture of deformable parts: from Felzenszwalb et al. [2010] define a mixture of models, where each model is a star-based pictorial structure. This can be achieved by restricting the co-occurrence model to allow for only globally-consistent types:

$$b_{ij}^{t_i,t_j} = \begin{cases} 0 & \text{if } t_i = t_j \\ -\infty & \text{Otherwise} \end{cases} \tag{3.3}$$

Articulation: the author also propose a simplified version of 3.2 with a reduced set of springs:

$$\omega_{ij}^{t_i,t_j} = \omega_{ij}^{t_i} \tag{3.4}$$

Inference

Inference corresponds to maximizing $S(I, l, t)$ from 3.2 over l and t. When the relational graph $G = (V, E)$ is a tree, this can be done efficiently with dynamic programming. To illustrate inference, let us re-write 3.2 by defining $z_i = (l_i, t_i)$ to denote both the discrete pixel location and discrete mixture type of part i:

$$\begin{aligned} S(I,z) &= \sum_{i \in V} \phi_i(I, z_i) + \sum_{ij \in E} \psi_i j(z_i, z_j), \\ where \quad \phi_i(I, z_i) &= \omega_i^{t_i} \cdot \phi(I, l_i) + b_i^{t_i} \\ \psi_{ij}(z_i, z_j) &= \omega_{ij}^{t_i,t_j} \cdot \psi(l_i - l_j) + b_{ij}^{t_i,t_j} \end{aligned} \tag{3.5}$$

From this perspective, it is clear that our final model is a discrete, pairwise Markov random field (MRF). When $G = (V, E)$ is tree-structure, one can compute $max_z\, S(I, z)$ with dynamic programming.

To be precise, we iterate over all parts starting from the leaves and moving "upstream" to the root part. We define kids(i) be the set of children of part i, which is the empty set for leaf parts. We compute the message part i passes to its parent j by the following:

$$score_i(z_i) = \phi_i(I, z_i) + \sum_{k \in kids(i)} m_k(z_i) \tag{3.6}$$

$$m_i(z_j) = \max_{z_i} [score_i(z_i) + \psi_{ij}(z_i, z_j)] \tag{3.7}$$

3.6 computes the local score of part i, at all pixel locations l_i and for all possible types t_i, by collecting messages from the children of i. 3.7 computes for every location and possible type of part i, the best scoring location and type of its child part i. Once messages are passed to the root part ($i = 1$), $score_1(z_1)$ represents the best scoring configuration for each root position and type. One can use these root scores to generate multiple detections in image I by thresholding them and applying non-maximum suppression (NMS). by keeping track of the $argmax$ indices, one can backtrack to find the location and type of each part in each maximal configuration. To find multiple detections anchored at the same root, one can use $N - best$ extensions of dynamic programming. *Computation:* The computationally taxing portion of dynamic programming is 3.7. We rewrite this step in detail:

$$m_i(t_i, l_j) = \max_{t_i} \left[b_{ij}^{t_i,t_j} + \max_{l_i} score_i(t_i, l_i) + \omega_{ij}^{t_i,t_j} \cdot \psi(l_i, l_j) \right] \tag{3.8}$$

one has to loop over LxT possible parent locations and types, and compute a max over LxT possible child locations and types, making the computation $O(L^2T^2)$ for each part. When $psi(l_i - l_j)$ is a quadratic function (as is the case for us), the inner maximization in 3.8 can be efficiently computed for each combination of t_i and t_j in $O(L)$ with a max-convolution or distance transform [Felzenszwalb and Huttenlocher, 2005]. Since one has to perform T^2 distance transforms, message passing reduces to $O(LT^2)$ per part.

Learning

We assume a supervised learning paradigm. Given labeled positive examples $\{I_n, l_n, t_n\}$ and negative examples $\{I_n\}$, we will define a structured prediction objective function similar. To do so, let us write $l_n = (l_n, t_n)$ and note that the scoring function 3.2 is linear in model parameters $\beta = (\omega, b)$, and so can be written as $S(I, z) = \beta \cdot \Phi(I, z)$. We would learn a model of the form:

$$\begin{aligned} arg \quad & mim_{\omega, \xi_n \geq 0} \quad \frac{1}{2}\beta \cdot \beta + C \sum_n \xi_n \\ s.t. \forall n \in pos \quad & \beta \cdot \Phi(I_n, z_n) \geq 1 - \xi_n \\ \forall n \in neg, \forall z \quad & \beta \cdot \Phi(I_n, z) \leq -1 + \xi_n \end{aligned} \tag{3.9}$$

The above constraint states that positive examples should score better than 1 (the margin), while negative examples, for all configurations of part positions and types, should score less than -1. The objective function penalizes violations of these constraints using slack variables ξ_n.

Dual coordinate descent: The currently fastest solver for linear SVMs appears to be liblinear, which is a dual coordinate descent method. A naive implementation of a dual SVM solver would require maintaining a MxM kernel matrix, where M is the total number of active constraints (support vectors). The innovation of liblinear is the realization that one can implicitly represent the kernel matrix for linear SVMs by maintaining the primal weight vector β, which is typically *much* smaller. In practice, dual coordinate descent methods are efficient enough to reach near-optimal solutions in a simple pass through large datasets.

In practice, to apply this algorithm, We define parts to be located at joints, so these provide part position labels l, but not part type labels t. further dataset detail can be found in chapter 4

3.2.2. Template based Model

This technique has played an important role in tracking-by-detection applications for many years. It is usually better adapted to low textured objects than feature point approaches. Unfortunately, this increased robustness often comes at the cost of increased computational load that makes direct template matching inappropriate for real -time applications. This is especially true as many templates must be used to cover the range of possible viewpoints. for our application due of physical constraint of the camera rig system and the design of the aquaculture infrastructure the viewpoints to the tracking object (Fish) is always coming from the side. from this viewpoints the amount of template necessary to have a proper result is reduce, as we explain at the beginning of this chapter.

Our approach is based on the recent and efficient template matching methods from Hinterstoisser et al. [2011, 2012] which, for our approach, consider only the images and their gradients to detect objects, because we not count with 3D data information. As such, they work even when the object is not textured enough to use feature point techniques, and learn new object virtually instantaneously. we based our solution in this algorithm, because has demostrated a great perfomance in presence of strong background clutter and due the prior knowledge of the object can handle in a better way occlusions, as the ones shows in the fig 3.7.

Also the algorithm show being much faster for larger templates set. Instead of making the templates invariant to small deformations and translation by considering dominant orientation only, the build a representation of the input images which has similar invariance properties but consider all gradient orientations in local image neighbourhood. Together with a novel similarity measure, this prevents problems due to too strong gradients in the background.

This algorithm consider the modern CPU architecture through exploit heavy SSE parallelization, structuring the the images representation to avoid “memory cache misses” which slow down the computations.

Model

In this section, we describe the template representation and show how the representation of the input image can be use to parse the image to quickly find objects. We will start by

Figure 3.6.: Fishes in aquaculture cage, heavily clutter background and occlusions

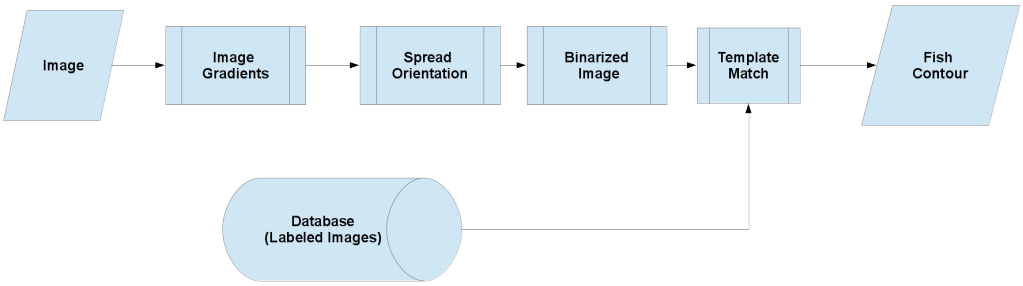


Figure 3.7.: Simplified flowchart of template based algorithm "'LINEMOD" Hinterstoisser et al. [2011]

deriving the similarity measure, emphasizing the contribution of each aspect of it, it show how is implement the approach to efficiently use modern processor architectures.

Similarity Measure

The unoptimized similarity measure can be seen as the measure defined by Steger [2002] modified to be robust to small translations and deformations.

$$\varepsilon_{steger}(I, \tau, c) = \sum_{r \in P} |cos(ori(\Theta, r) - ori(I, c + r))| \tag{3.10}$$

where $ori(\Theta, r)$ is the gradient orientation in radians at locations r in a reference image Θ of an object to detect. Similarly, $ori(I, c + r)$ is the gradient orientation at c shifted by r in the input image I. We use a list, denoted by P, to define the locations r to be considered in Θ. This way we can deal with arbitrarily shaped objects efficiently. A template τ is therefore defined as a pair $\tau = (\Theta, P)$.

Each template tau is created by extracting a small set of its most discriminant gradient orientations from the corresponding reference image as shown in figure 3.8 and by storing their locations. To extract the most discriminative gradients we consider the strength of their norms. In this selection process, we also take the location of the gradients into account to avoid an accumulation of gradient orientations in one local area of the object while the rest of the object is not sufficiently described.

Considering only the gradient orientations and not their norms makes the measure robust to contrast changes, and taking the absolute value of the cosine allows it to correctly handle object occluding boundaries: It will not be affected if the object is over a dark background, or a bright background.

The similarity measure of Eq. 3.10 is very robust to background clutter, but not to small shifts and deformations. A common solution is to first quantize the orientations and to use local histograms like in SIFT Lowe [2004] or HOG Dalal and Triggs [2005]. However this can be unstable when strong gradients appear in the background.

To overcome this issues, Hinterstoisser et al. [2011] introduce a similarity measure that, for each gradient orientation on the object, searches in a neighbourhood of the associated gradient location for the most similar orientation in the input image. This can be formalized as:

$$\varepsilon(I,\tau,c) = \sum_{r\in P}\left(\max_{t\in R(c+r)} |cos(ori(\Theta,r) - ori(I,t))|\right) \tag{3.11}$$

where $R(c+r) = [c+r-\frac{T}{2}, c+r+\frac{T}{2}]\, X\, [c+r-\frac{T}{2}, c+r+\frac{T}{2}]$ defines the neighbourhood of size T cantered on location $c+r$ in the input image. Thus, for each gradient it is align the local neighbourhood exactly to the associated location whereas in DOT, the gradient orientation is adjusted only to some regular grid. We show below how to compute this measure efficiently.

Computing the Gradient Orientations

Before to continue with the background, we shortly discuss why we use gradient orientations and how we extract them easily. The chose to consider image gradients because they proved to be more discriminant than other forms of representations, and are robust to illumination change and noise. Additionally, image gradients are often the only reliable image cue when it comes to texture-less objects. Considering only the orientation of the gradients and not their norms makes the measure robust to contrast changes, and taking the absolute value of cosine between them allows it to correctly handle object occluding boundaries: it will not be affected if the object is over a dark background or a bright background. To increase robustness, this algorithm compute the orientation of the gradients on each color channel of our input image separately and for each image location use the gradient orientation of the channel whose magnitude is largest. Given a RGB color image I, We compute the gradient orientation map I_g at location x with

$$I_g(x) = ori(\hat{C}(x)) \tag{3.12}$$

where

$$\hat{C}(x) = \arg \max_{C \in R,G,B} \left\| \frac{\partial C}{\partial x} \right\| \tag{3.13}$$

and R, G, B are the RGB channels of the corresponding color image.

In order to quantize the gradient orientation map, the gradient direction is omitted, consider only the gradient orientation and divide the orientation space into n_0 equal spacing as shown in figure 3.8. To make the quantization robust to noise, we assign to each location the gradient whose quantized orientation occurs most often in a $3x3$ neighbourhood. We also keep only the gradients whose norms are larger than a small threshold.

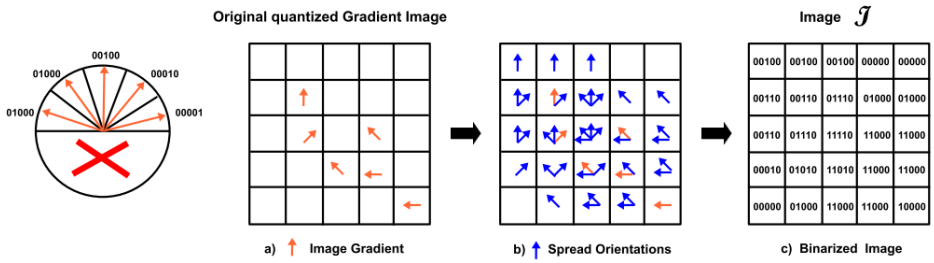


Figure 3.8.: Spreading the gradient orientations. Left: The gradient orientations and their binary code. We do not consider the direction of the gradients. a) The gradient orientations in the input image, shown in orange, are first extracted and quantized. b) Then, the locations around each orientation are also labeled with this orientation, as shown by the blue arrows. This allows our similarity measure to be robust to small translations and deformations. c) is an efficient representation of the orientations after this operation, and can be computed very quickly.

3.3. Proposed Workflow

The aim of this section is explain our workflow to achieve the goal define in chapter 1, and addresses the three major challenges, which are: *Deformations, Different Viewpoints, Oclussions*.

The propose workflow, as shown in figure 3.9, consists of two main stages, First, The Part Based model algorithm presented in section 3.2.1, which has a a output a bounding box associate to a probability to this part of the image correspond a fish part, as the are defined in the corresponding labeled dataset. And second, using the output from the first algorithm to apply the template matching only on the areas define by the bounding box, in this way is easier to get rid off the outliers. In this point is important to explain the consideration regarding the use of linemod in a **non-rigid** object detection.

As we explain in section 3.0.1; due to the physiological properties of our target, we can model it as a two part model, Head and Tail, and assuming that those parts are rigid enough to consider them as rigid objects separately. as we already know from the previous stage, which bounding box belong to which candidate fish, we just need to check that the

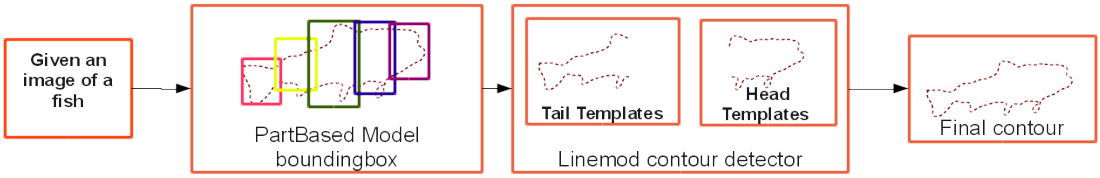


Figure 3.9.: Fish contour extraction - Propose workflow

contour found for head and tail fit between each other. to accomplish this task we extract from the detected contour the following parameter:

- *found a ellipse*, we calculate the ellipse that fits (in a least-squares sense) the set of 2D points define by the contour.
- *connector points*, We also extract from the contours, the so call, connector point which correspond to the to extreme of the contour as show in figure 3.10

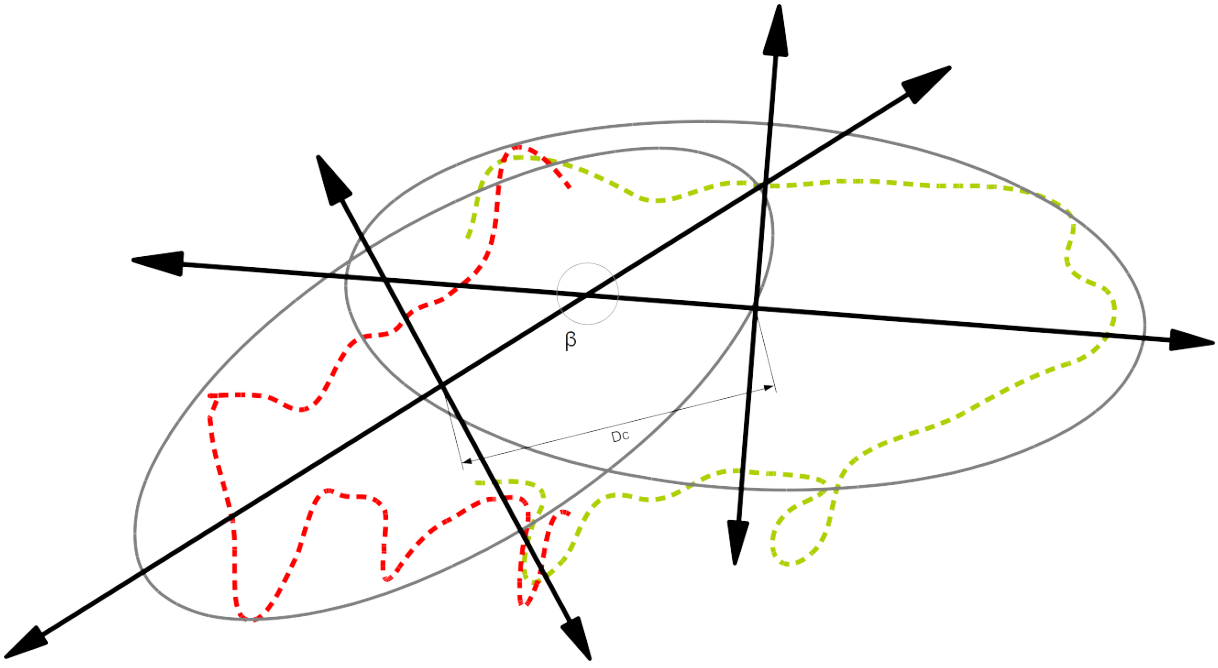


Figure 3.10.: This illustrate the final check, which is perform to the detected contours

With those two parameters, then we can check if the correspond found contours are close enough and in the right position to form a the fish contour. A *pseudo-code* is presented in algorithm 1 to explain our propose workflow.

input : A new Image Im of the fishes
input : Trained detector $Td_{pbmFish}$ Part Based model
input : Trained detector Td_{tmHead} Template matching for Head
input : Trained detector Td_{tmTail} Template matching for Tail
output: Fish contour

;
fishCandidates← getCanditeFromPartBasedModel(Im, $Td_{pbmFish}$);
foreach *fishCandidate in fishCandidates* **do**
    Go through all fish candidates and apply template matching;
    headContours ← contourbyLinemod(I_m, Td_{tmHead}, HeadBoundingBox);
    tailContours ← contourbyLinemod(I_m, Td_{tmTail}, TailBoundingBox);
    foreach *headContour in headContours* **do**
        Go through all head contour and check if the are connected with the tail;
        foreach *tailContour in tailContours* **do**
            if *tailContour connected with headContour* **then**
                Candidate ← is Fish
            end
            else
                continue
            end
        end
    end
end

Algorithm 1: Propose workflow for fish contour detection

4. Implementation

In this chapter, we discuss implementation details of our approach introduced in 3. The first section discusses the creation of a annotated dataset for train, separately, the part based detection and the template based algorithm, while in the second section, we describe will describe development environment, and training procedure.

4.1. Dataset - Part based model

In this section, We discuss the details involved in the creation of the dataset for training the algorithm defined in chapter 3, during the project we acquire a big amount of data, considering different in illumination, water properties and with different fish types, but for this work, and due that the data preparation is handwork intensely, we process a specific set of data, in which all the experiments will be performed, to achieve this goal, a small C++ application was developed. As introduced in 3 the fish is model using the concept of stickmen, that mean, each part is define by its coordinates (x, y) and joint through a line. analysing the fish structure, we create two dataset modelling the fish with 3 and 7 parts, as shown in figure 4.1 and 4.1 respectively.

base on this two created dataset, it is possible then to extrapolate the model linearly, using a simple matrix multiplication as define in equation 4.4, enabling us to test our propose algorithm with for example 5, 9, 11 parts fish model.

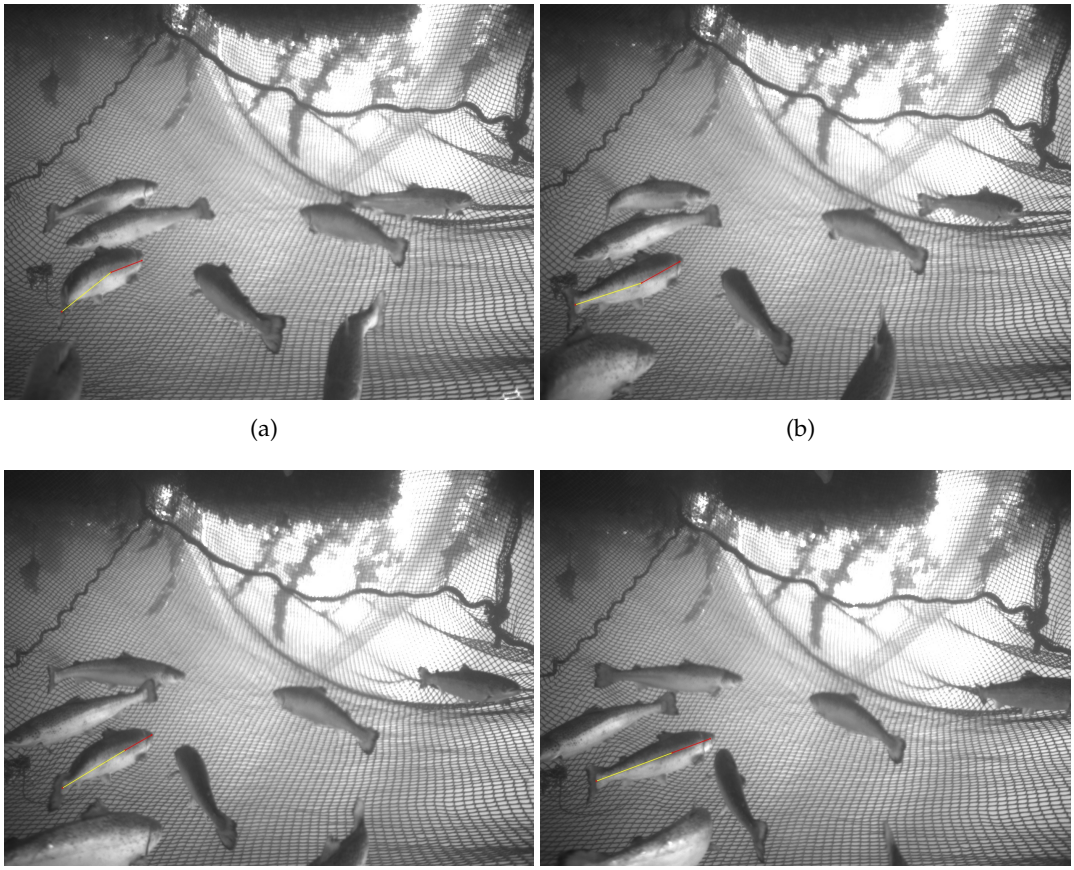
$$\hat{point}_{m'xn} = \mathtt{A}_{m'xm} \cdot point_{mxn} \tag{4.1}$$

where

$$\mathtt{A} = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & \ddots & & & & \vdots \\ 0 & a_{32} & a_{33} & a_{34} & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & a_{(m'-2)(m-3)} & a_{(m'-2)(m-2)} & a_{(m'-2)(m-1)} & 0 \\ \vdots & & & & \ddots & a_{(m'-1)(m-2)} & a_{(m'-1)(m_1)} & a_{(m'-1)m} \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_{m'(m-1)} & a_{m'm} \end{bmatrix} \tag{4.2}$$

now as a example, to extrapolate from a 3 parts model to a 5 parts model, the matrix should look like this:

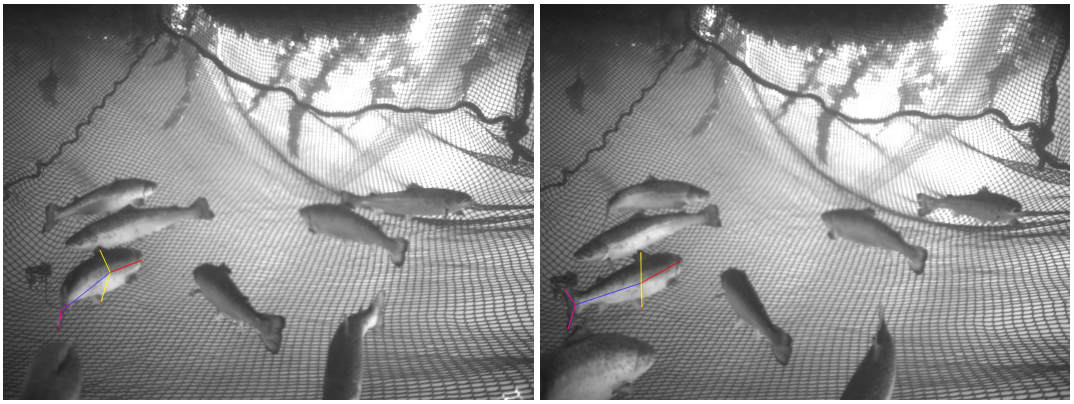
$$\hat{point}_{9x2} = \mathtt{A}_{9x3} \cdot point_{3x2} \tag{4.3}$$



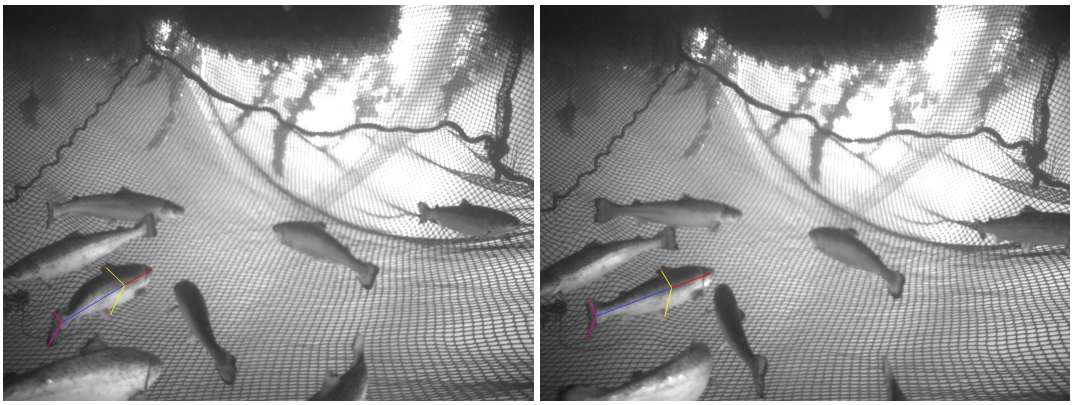
(a) (b)

(c) (d)

Figure 4.1.: This illustrates the annotated fish in a 3 Part model

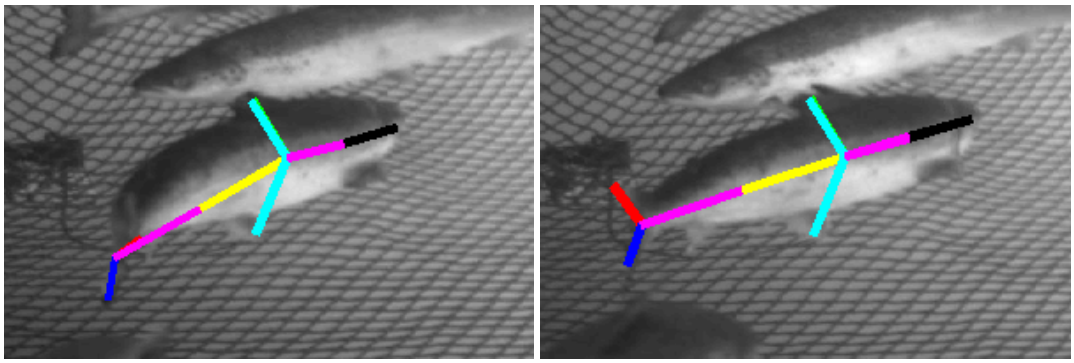


(a) (b)



(c) (d)

Figure 4.2.: This illustrates the annotated fish in a 7 Part model



(a) (b)

Figure 4.3.: This illustrates the annotated fish in a 9 Part model, where is clearly visible the new points define in the middle point of the existent parts.

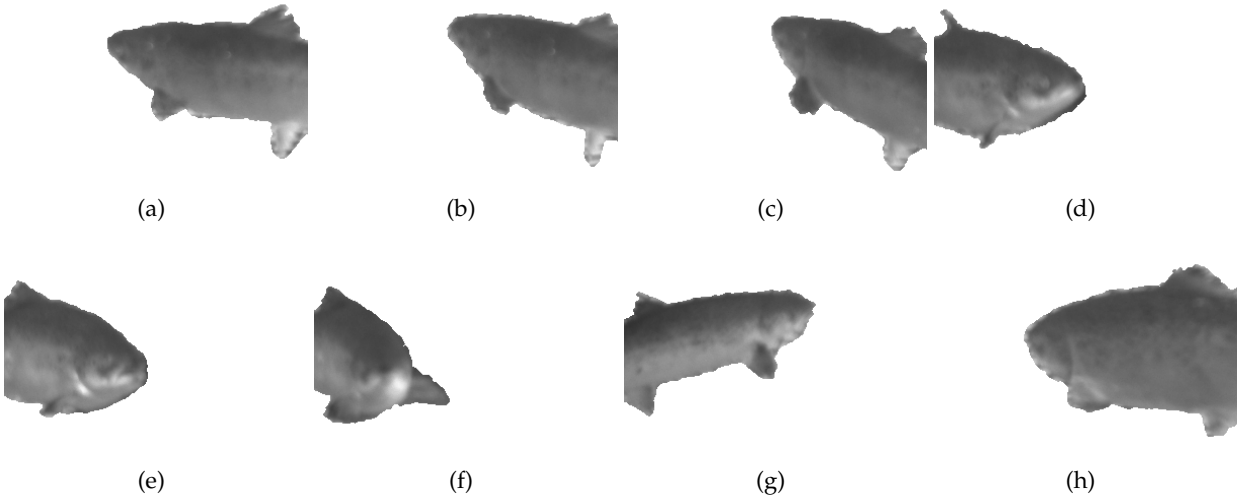
where

$$\mathtt{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

and each $\frac{1}{2}$ represent the middle point between the two original annotated points.

4.2. Dataset - Template based model based model

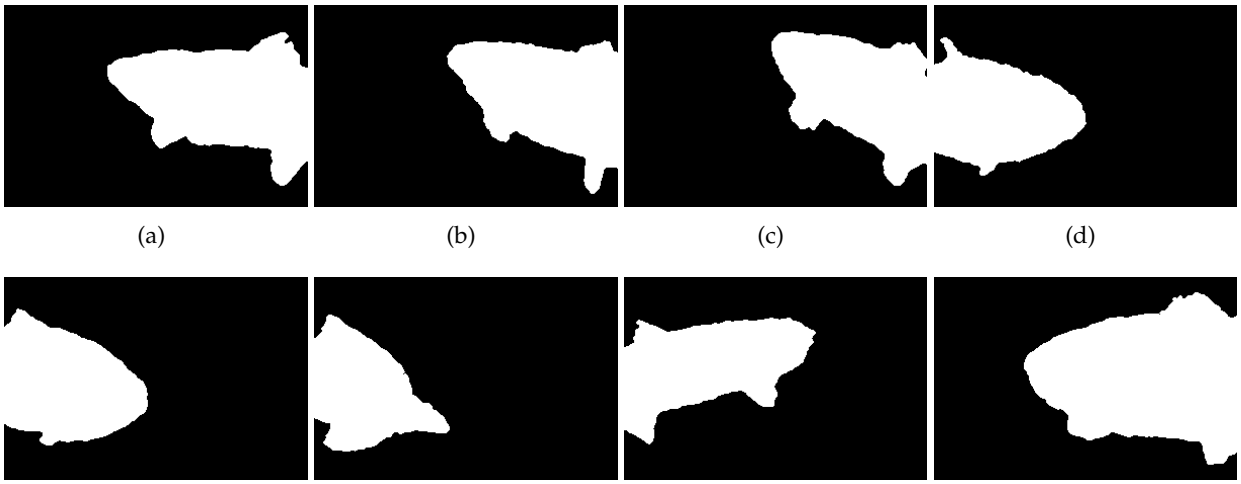
In this section, We discuss the details involved in the creation of the dataset for training the algorithm defined in 3.2.2, as said before for the template matching algorithm, we propose a different way to generate the proper templates, we have template for the Head and Tail, and we train two different linemod detector, one for each part, in the figures 4.2, 4.2, 4.2 and 4.2 is show the head of the fish and his corresponding annotated mask, the same can be appreciated for the tail of the fish. the task of create the dataset for this project, was one of the most time consuming part, due to, as mentioned before, the nature of the linemod algorithm is to be apply on rigid object, and we are approximating to this assumption by dividing the fish in two part, expecting that the deformation suffer from the fish, can be isolate into those two part.



(a) (b) (c) (d)

(e) (f) (g) (h)

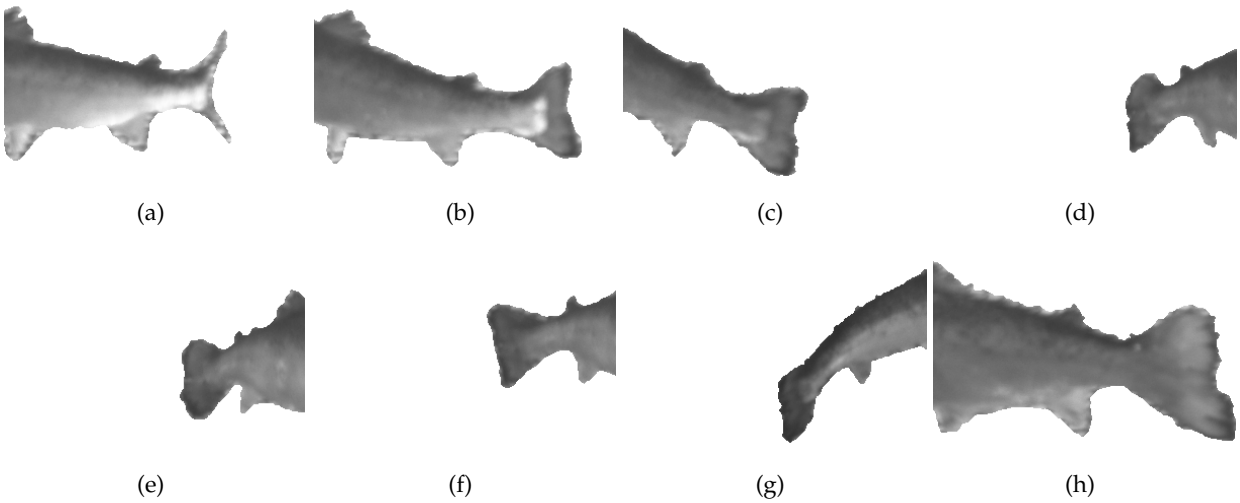
Figure 4.4.: This illustrates the annotated fish head, need it for train the linemod detector



(a) (b) (c) (d)

(e) (f) (g) (h)

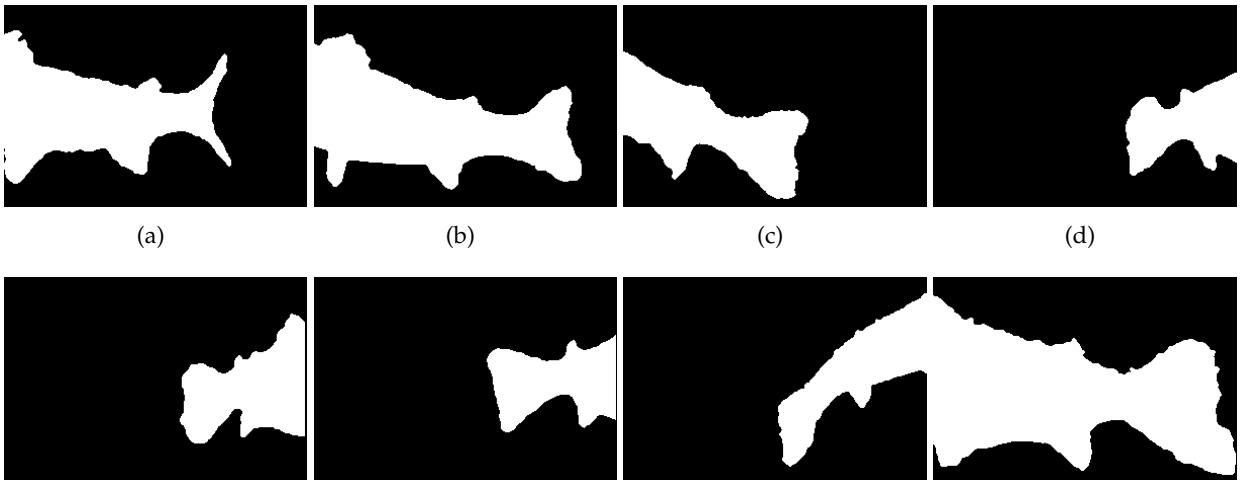
Figure 4.5.: This illustrates the annotated mask fish head, need it for train the linemod detector



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.6.: This illustrates the annotated fish tail, need it for train the linemod detector



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.7.: This illustrates the annotated mask fish tail, need it for train the linemod detector

4.3. Software implementation

In this section, we will discuss implementation details involve in this work, relate with tools and algorithm used.

- *Part Based algorithm* from section 3.2.1 was implemented in MATLAB, and the code can be found on the author website (http://www.ics.uci.edu/~dramanan/), this code was modified to process our specific dataset. and considering the highly expensive computational require by this algorithm, the deployment of it was perform in the *Linux-Cluster* from our University. which consists of several segments with different types of interconnect and different sizes of shared memory. All systems have a (virtual) 64 bit address space.
  - *Learning* our learning process starts by loading the 2D intensity image from the annotated dataset, and randomly sampling the images and pixels. this samples are used as input as define for equation 3.9. This process is repeated until all the constraint are fulfilled, and the learned parameter are save as a MAT file (Microsoft Access Table Shortcut file).
  - *Prediction and Runtime* Using a different dataset, we start loading the trained model, given a 2D image the algorithm output a set of candidates, define by its bounding boxes, to be able to combine the template based algorithm, which is implemented in C++ and using opencv, and this one, we used a implementation of the algorithm ported to C++ and is available in (https://github.com/wg-perception/PartsBasedDetector).
- Template based algorithm from section **??** was implemented in C++ using the computer vision library OpenCV.
  - *Learning* our learning process starts by loading the 2D intensity image from the annotated dataset, recall that two different detector are trained, one for the head and other for the tail, and Adding the template to the database as mention in Hinterstoisser et al. [2012] and describe in section **??**. This process in repeated to our entire dataset and learned parameter are save as a binary file.
  - *Detection and Runtime* Using a different dataset, we start loading the trained model, given a 2D image with its corresponding labeled mask, coming from the part based algorithm, the algorithm output a set of detect contour for the head and tail of the fish, Initially, the output was not as good as expected, due to the nature of the training data, but then, we include a verification test to ensure that the set of head-tail contour is geometrically feasible, as explained in section 3.3

Part III.

Results and Conclusion

5. Results and Discussion

5.1. Results and Discussion

In this chapter, we evaluate the proposed algorithm using real data acquire using the camera rig system illustrated in **??** and approaches

define approaches

5.2. Datasets

5.2.1. Part based model

We define, as describe in chapter 4, a full-body skeleton for the fish annotated dataset, with this information we construct a fully supervised dataset, from which we learn a flexible mixture of parts, We show the full-body model learned from the created data set, for 3, 5, 7, 9 and 11 part fish model. those are presented in figure , we set all part to be 5 x 5 HOG cells in size. to visualize the model, we show 4 tress generated by selecting one of the four types of each part. and placing it at its maximum-scoring position. Recall that each part has its own appearance template and spring encoding its relative location with respect to its parent.

reffig::::::

in table ref

Evaluation Criteria

In this section we describe the evaluation criteria used to evaluate pose estimation. The probability of correct pose (PCP), broadly-adopted evaluation protocol measures the percentage of correctly localized body parts. A candidate body part is labeled as correct if its segment endpoint lie in 50% of the length of the ground-truth anottated endpoints. the result of this evaluation criteria are show in the table

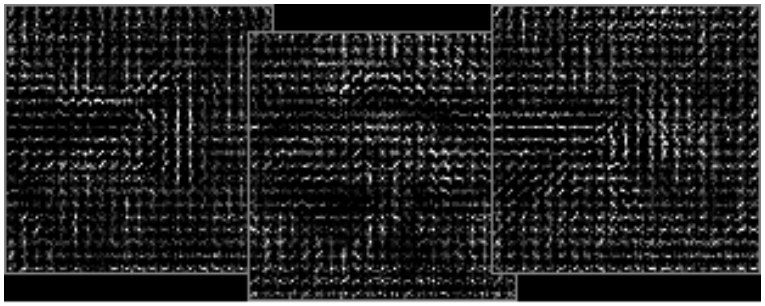
reftable

organize this section

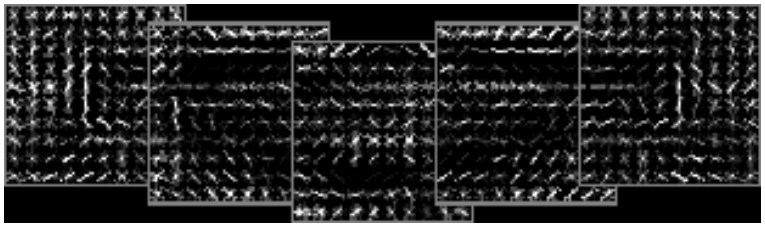
Number of mixture parts	PCP
3	
5	
7	

Table 5.1.: We evaluate various strategies for trainings parts.

bring the table from the presentation

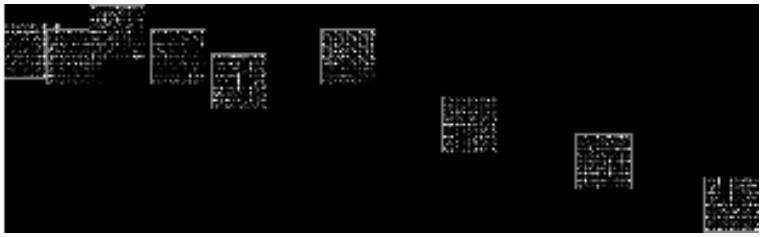


(a) K=3



(b) K=5

(c) K=7



(d) K=9

(e) K=11

Figure 5.1.: This figure illustrate our 3, 5, 7, 9, 11 part model, demonstrate that the additional parts up to 7 part, doesn't increase performance

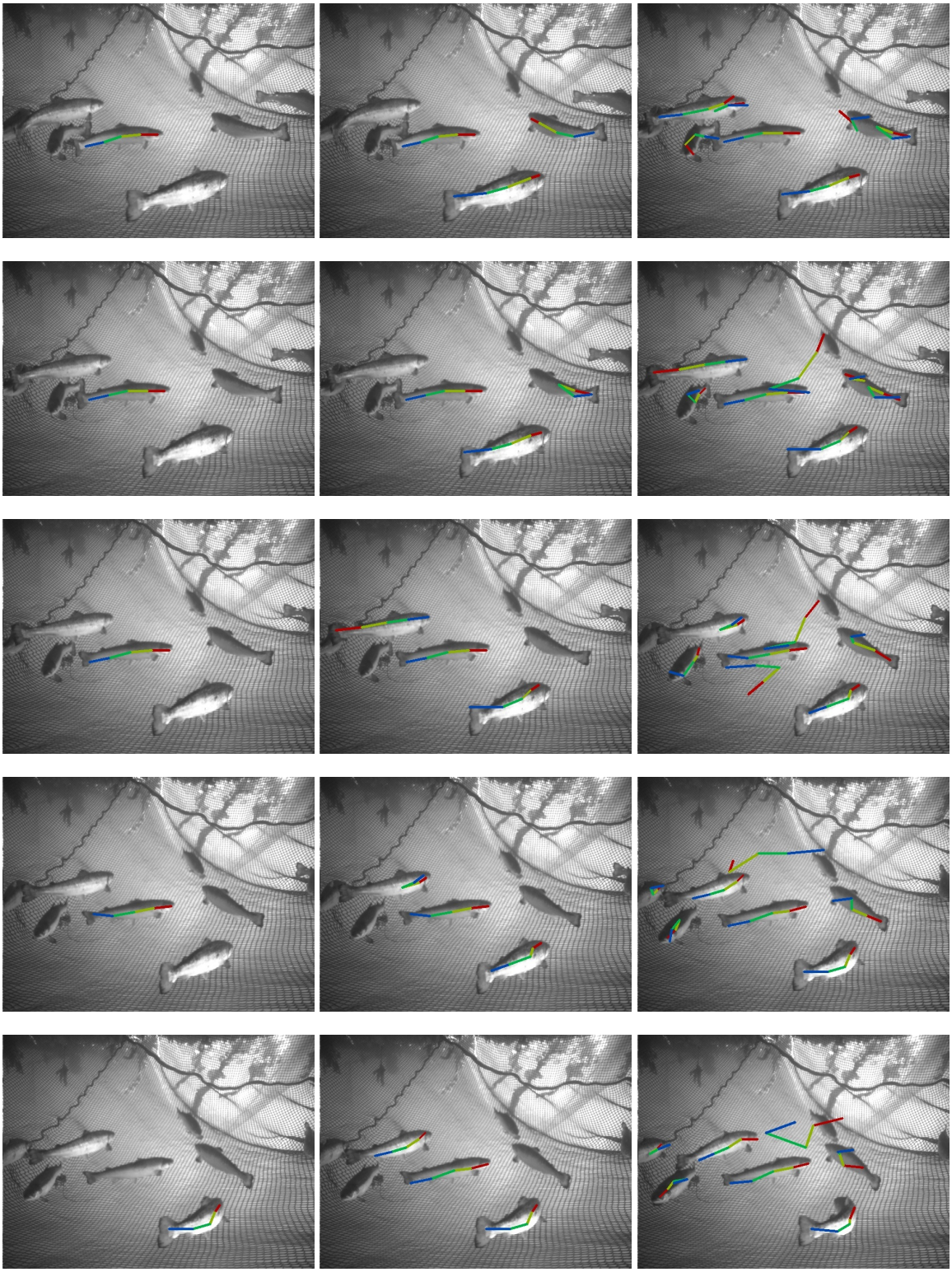


Figure 5.2.: 5 best, 3, 7

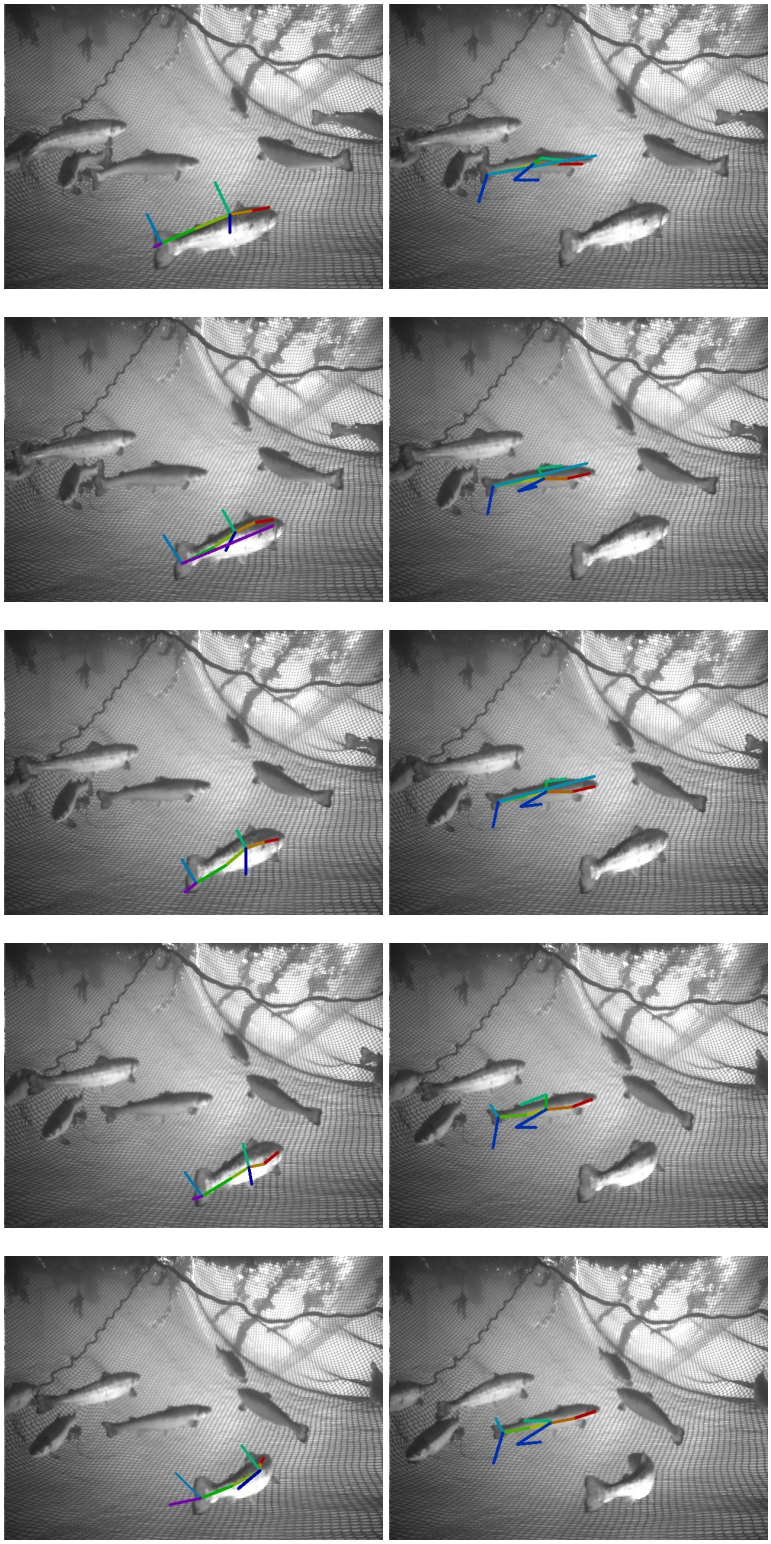
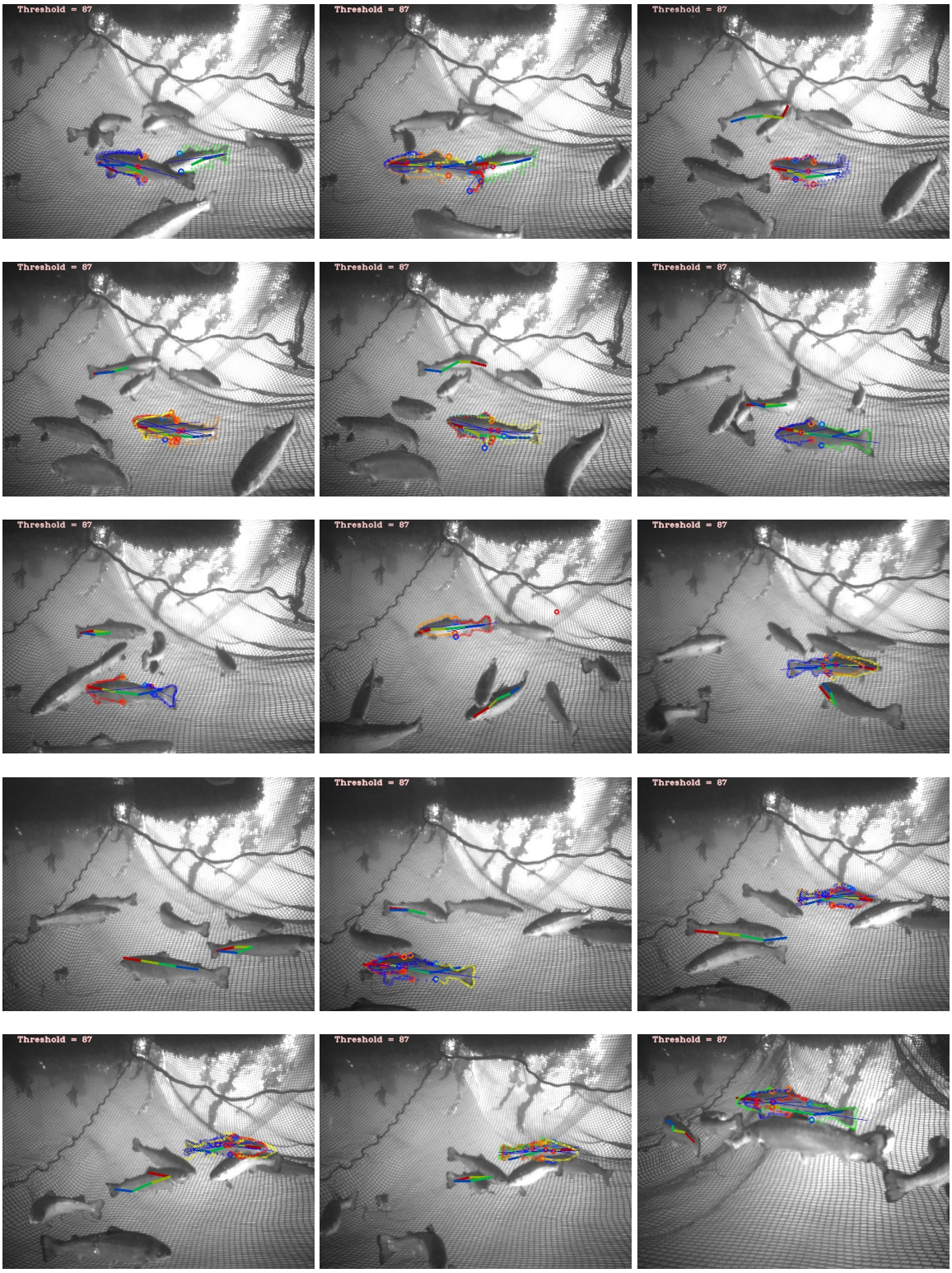


Figure 5.3.: my caption


Threshold = 87

finish the discussion

5.3. Discussion

6. Conclusion

6.1. Conclusion

In this thesis, we accomplished the goals of creating a annotated datasets that comprise of learning keypoints and fish contours from a set of 2D grayscale images. We also implement a combine solution from a part based detection model and a template matching detection approach, which is capable of predict keypoints and fish contours in an unlabeled 2d grayscale image and verifying the validity of the prediction.

Inspired on the works from **?** and Hinterstoisser et al. [2012], we use the part based model Based on the results, we conclude that the algorithm work best when the sagittal plane of the fish is parallel to the image plane. In addition, we also conclude

Lastly, we demonstrate our algorithm on multiple fish detection by

more detail about the algorithm

.....

.....

6.2. Future Work

Multiple fish detection and occlusions are some of the biggest challenges in this project and needs to be addressed because they are always present in real situations. Another important task for the future would be to increase the size of the labeled database to improve the detection result, and perform a benchmarking of other approaches to have a better picture of what are the challenges involve in this problem.

Since the final product combine a Time of flight camera and a 2D HD grayscale camera in a stereo system rig inside a underwater housing, the next step would be use the strength from each device into fusing the depth information with the 2D intensity image. This present new challenges as the system is design with underwater purpose. the new approach should deal with the systematical error introduce in perspective pinhole camera mode by several refraction of the light rays causes by the housing configuration, due to the glass interface between water and air or other inert gas introduce in the housing, this issue could be handle by the approach propose in this work Sedlazeck and Koch [2011].

After accomplish a proper data fusion between the two modalities, the next step would be find a algorithm capable of take advantage of all this information in a optimize way.

Appendix

A. Detailed Descriptions

Here come the details that are not supposed to be in the regular text.

Bibliography

Frank Asche and Trond Bjorndal. *The Economics of Salmon Aquaculture, Second Edition*. Wiley-Blackwell, 2011.

Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

J.E. Colgate and K.M. Lynch. Mechanics and Control of Swimming: A Review. *IEEE Journal of Oceanic Engineering*, 29(3):660–673, July 2004. ISSN 0364-9059. doi: 10.1109/JOE.2004.833208. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1353419.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, volume I, pages 886–893, 2005.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

Elliot Hawkes, Chris Quinn, and Robert J. Wood. Design, fabrication and analysis of a body-caudal fin propulsion system for a microrobotic fish. *2008 IEEE International Conference on Robotics and Automation*, pages 706–711, May 2008. doi: 10.1109/ROBOT.2008.4543288. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4543288.

Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. *2011 International Conference on Computer Vision*, pages 858–865, November 2011. doi: 10.1109/ICCV.2011.6126326. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6126326.

Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE transactions on pattern analysis and machine intelligence*, 34(5):876–88, May 2012. ISSN 1939-3539. doi: 10.1109/TPAMI.2011.206. URL http://www.ncbi.nlm.nih.gov/pubmed/22442120.

THOMAS A. LARSEN and FRANK ASCHE. Contracts in the Salmon Aquaculture Industry: An Analysis of Norwegian Salmon Exports, 2011. ISSN 0738-1360.

Yajie Liu, Jon Olaf Olaussen, and Anders Skonhoft. Wild and farmed salmon in Norway-A review. *Marine Policy*, 35(3):413–418, 2011.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Arif Mahmood and Sohaib Khan. Correlation-coefficient-based fast template matching through partial Elimination. *IEEE Transactions on Image Processing*, 21(4):2099–2108, 2012.

Anne Sedlazeck and Reinhard Koch. Calibration of Housing Parameters for Underwater Stereo-Camera Rigs. *BMVC*, 2011. URL http://www.bmva.org/bmvc/2011/proceedings/paper118/paper118.pdf.

Mark R Shortis, Mehdi Ravanbakskh, Faisal Shafait, Philip Culverhouse, Danelle Cline, and Duane Edgington. A review of techniques for the identification and measurement of fish in underwater stereo-video image sequences. *SPIE Optical ...*, 2013. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1691542.

Concetto Spampinato and YH Chen-Burger. Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos. *VISAPP (2)*, 2008. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.63.7796&rep=rep1&type=pdf.

Carsten Steger. Occlusion, clutter, and illumination invariant object recognition. *... Archives of Photogrammetry Remote Sensing and ...*, 2002. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:OCCLUSION+,+CLUTTER+,+AND+ILLUMINATION+INVARIANT+OBJECT+RECOGNITION#0.

Robin Tillett, Nigel Mcfarlane, and Jeff Lines. Estimating Dimensions of Free-Swimming Fish Using 3D Point Distribution Models. *Computer Vision and Image Understanding*, 79: 123–141, 2000. ISSN 10773142. doi: 10.1006/cviu.2000.0847.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1, 2001.

Yi Yang and Deva Ramanan. Articulated Human Detection with. pages 1–15.