NTNU – Trondheim
Norwegian University of
Science and Technology

Evaluation of Methods for Robust, Automatic Detection of Net Tear with Remotely Operated Vehicle and Remote Sensing

Tormod Haugene

Master of Science in Cybernetics and Robotics
Submission date: June 2014
Supervisor: Tor Arne Johansen, ITK
Co-supervisor: Eirik Svendsen, SINTEF Fiskeri og Havbruk

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Preface

This thesis concludes my master studies at the Norwegian University of Science and Technology's (NTNU) Department of Engineering Cybernetics. The project was a collaboration between NTNU and The Society for Industrial and Technological Research Fisheries and Aquaculture (SINTEF F&A).

I would like to thank NTNU and SINTEF F&A for this unique opportunity to utilize me theoretical knowledge in a practical manner in what may be considered a progressive field of research. Through this thesis, I have gained knowledge of the multiple challenges of applying computer vision technology within the aquaculture fish farming industry, as well as the usage automation systems in an underwater environment.

I would especially like to thank my two main supervisors Eirik Svendsen, at SINTEF F&A, and Tor Arne Johansen, at NTNU, for their constant support and advice during this project. Furthermore, I highly appreciate that John Reidar Mathiassen and Per Rundtop at SINTEF F&A took their time to follow the progression of this thesis, and assist with solving theoretical questions encountered. Finally, my gratitude goes to SINTEF F&A, Morten Sølvberg Sletta and Morten Engelhardt Olsen for providing of the video material analyzed throughout this thesis.

Trondheim, 2014-06-09

Tormod Haugene

Tormod Haugene

.

Abstract

Accompanying the continuous growth of the aquaculture fish farming industry in the recent years, the usage of Remotely Operated Vehicles (ROV) for regular inspections of net integrity has become increasingly common. For a human ROV operator, routine inspections can be repetitious and time consuming, and improving the regularity and efficiency of these operations are of interest. The aim of this study was therefore be to develop a robust technique for automatic detection of net damage with an ROV mounted camera and computer vision, which later can be employed either as an aid for a human operator or be embedded into an automatic solution in the future. Information from temporal background segmentation, edge detection, motion estimation and multiple image channels was incorporated into a high-redundancy combinatorial system design for background segmentation. Assessment of net damage was made from the resulting binary foreground image by employing a detection scheme based on morphological operations. The background segmentation performance, detection accuracy and robustness of the developed system was evaluated on previously recorded video material from real ROV operations and a simulated test setup. Results showed that the background segmentation process provided a stable and comprehensive binary foreground image, but with reduced ability to segment certain foreground objects. The damage assessment methodology, on the other hand, displayed a rigorous evaluation capability. With some additional measures, the developed procedure seems promising for achieving robust net damage detection in a practical implementation.

.

Sammendrag

Som følger av en kontinuerlig vekst innenfor havbruksnæringen de siste årene, har bruken av fjernstyrte undervannsfarkoster (ROV) for inspeksjon av nettintegritet vært økende. For en menneskelig operatør kan slike rutineoperasjoner være monotone og tidkrevende, og å forbedre regulariteten og effektiviteten til disse operasjonene er av interesse. Målet med denne masteroppgaven var derfor å utvikle en robust metode for automatisk deteksjon av nettskade ved hjelp av et kamera montert på en ROV og datasyn, en løsning som senere kan benyttes enten som et hjelpemiddel for en menneskelig ROV-operatør eller bli innebygget i et automatisk system i framtiden. Informasjon fra temporal bakgrunnssegmentering, kantgjenkjenning, bevegelsesestimering og flere bildekanaler ble sammenkoblet i et høyredundans kombinatorisk systemdesign for bakgrunnssegmentering. Undersøkelser av nettskade ble gjennomført basert på det resulterende binære forgrunnsbildet ved å benytte en deteksjonsprosess bestående av morfologiske operasjoner. Ytelsen, nøyaktigheten og robustheten til bakgrunnssegmenteringen ble evaluert ut fra tidligere oppsamlet videomateriell fra virkelige ROV-operasjoner og et simulert testoppsett. Resultatene viste at bakgrunnssegmenteringsprosessen gav et stabilt og helhetlig binært forgrunnsbilde, men med en redusert evne til å framstille enkelte forgrunnselementer. Metodikken til deteksjonsprosessen viste derimot solide evalueringsegenskaper. Gitt enkelte tiltak ser den utviklede metoden lovende ut for å oppnå robust deteksjon av nettskade i en praktisk implementasjon.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In 2009, products resulting from fish and fishery accounted for 16.6 percent of the world population's animal protein consumption, and 6.5 percent of its total protein intake (FAO, 2012). Over the past three decades (1980-2010), aquaculture production has expanded with an average rate of 8.8 percent annually, provisioning about 41 percent of all fish worldwide in 2011 (FAO, 2012). It is estimated that the production of salmonids from Norwegian aquaculture will increase fivefold between 2010 and 2050, given that environmental, political and technical prerequisites are met, calling for automated solutions and an environment preserving focus in research (Olafsen, Winther, Olsen, & Skjermo, 2012).

Prevention of fish escapes is a major environmental concern, as interbreeding between farmed fish and local fish stocks can alter the genetic material of the local stocks sufficiently to reduce its survivability in its natural habitat due to selective breeding in farmed fish (Fleming & Einum, 1997) (Hindar, Ryman, & Utter, 1991) (McGinnity et al., 1997). Between 2006 and 2009 in Norway, 68 percent of escapes of Atlantic salmon were found to be caused by structural and equipment related failures (Jensen et al., 2010). Similar results were observed in Scottish aquaculture, where 57 percent of fish escapes were found to originate from net damage between 2002 and 2009 (Taylor & Kelly, 2010). When the Norwegian technical standard for aquaculture (NS9415, 2009) first took effect in 2004, the total number of escapes of Atlantic salmon was drastically reduced (Jensen et al., 2010).

To uphold the Norwegian standard (NS9415, 2009), regular investigation of sea cage net integrity is required. Inspections are typically made before or after operations that may expose the cage structure to additional stress, such as delousing, fish delivery or mooring buoy maintenance, and on a regular, often monthly, basis. Net investigation is normally executed either by a team of (at least) three divers, or a Remotely Operated Vehicle (ROV) with a minimum of two on-scene operators (Arbeidstilsynet, 2011). Inspection of a sea cage with a 160 m circumference with good visibility typically requires about 20 minutes using divers, and about 60 minutes with an ROV. Detecting single mask net damages can be difficult even for an experienced inspector, in particular with presence of algae growth, while larger damages typically are found. In addition, Storvold states that post-operational inspection of the video feeds from the ROV-mounted camera sometimes may improve damage assessment over inspection with divers (personal communication, O. Krystad and B. M. Storvold, March 4, 2014).

The focus of this thesis will be on reducing the resources required during routine ROV inspections of sea cage net integrity by providing a computer vision (CV) tool to aid ROV-operators in identifying net damage from the ROV camera's video stream. The tool will attempt to detect net irregularities, and highlight them in the video stream, making them clearly visible to the ROV-operator, both during operation, and during post-operation video feed analysis. If this tool is successful, it may allow even less experienced ROV-operators to fully investigate a sea cage without post-operational video inspection, both while focusing on controlling the ROV itself, and without degrading the quality of the inspection. Later on, this tool might then be integrated into an Autonomous Underwater Vehicle (AUV) that automatically traverses the sea cage net wall, only signaling its operators if damage is detected or it has completed its operation and is ready for pickup. Although such a autonomous service might belong to the future of aquaculture technology, several steps of the process have already generated interest in certain research communities.

1.2 Previous Work

The idea of automating an underwater vehicle for doing sea cage net integrity inspection has been visited previously on a few occasions. No previous study has successfully tested a fully operational AUV for this purpose; however, individual features of a potential AUV has been studied. Master students from NTNU, in collaboration with SINTEF F&A, have been working on the subject through a series of projects and master theses, some of which are summarized here.

Automatic positioning and attitude estimation was tested by (Carlsen, 2010) on a low-cost, tethered mini-ROV by utilizing readings from an Inertial Navigation System (INS) implemented in a Kalman filter (Kalman, 1960). In addition, a simple path-following guidance system was developed, based on this technology. The path-following capability tests showed acceptable results, with increased positional deviation over time due to INS inaccuracy, as well as from pull from the tether cable at increasing depths (Carlsen, 2010).

A preliminary study of the potential of using an AUV in aquaculture sea cage inspections, was later considered by (Jakobsen, 2011). The study involved the development of a Graphical User Interface (GUI) for thruster control and video feed analysis for net damage detection using computer vision techniques with an ROV-mounted camera. Moreover, a laser module was utilized to estimate the distance between the ROV and the sea cage net wall, with accurate results. Jakobsen managed to track net meshes, given that: geometrical distortions from the camera optics were at a minimum; the net surface had little perspective distortion relative to the camera; the camera was at a range of between 15-60 cm from the net wall, depending on the mesh size of the net; and that the view of the net was clear of foreign obstacles, as these conflicted with the damage detection algorithm. Damage detection was attempted by first segmenting the net from the background with locally applied percentile thresholding on selected color channels, and then secondly traversing the resulting binary image with a line search algorithm based on depth first search (DFS). The background segmentation approach was compared to a method based on the Canny Edge Detector, combined with a method he named ”flooding” that connected isolated edges. Jakobsen concluded that separating the background with percentile thresholding gave the better result, and that checking net integrity by computer vision seemed promising, given further work on algorithms and improved ROV hardware (Jakobsen, 2011).

Using an improved hardware platform, two further studies were conducted on evaluating computer vision techniques to inspect sea cage nets. The writers collaborated in setting up an experiment from where video footage of different net damage types were collected (see Figure 1.4 (e)-(g)). While parts of their work was a joint effort, they produced separate reports with two different approaches to tracking the net - a technique that potentially could be utilized for assessing net damage:

The first thesis attempted to track the movement of the net wall based on the relative motion of objects in the video feed by applying various algorithms for optical flow. A classical algorithm based on phase correlation was concluded as the better approach of detecting the net's translation between frames, showing robustness to noise and a better ability to track dominant image structures. Also, tracking the net with Hough Transforms - line, probabilistic line, and circle - was tested, but without success (Olsen, 2013).

In the second thesis, by (Sletta, 2013), a variety of common computer vision techniques were explored, some of which were combined into workflows and used in combination with specifically developed damage and growth assessing algorithms. In particular, Sletta investigated the use of three thresholding techniques - Otsu's Method, Direct and Adaptive Thresholding - in comparison with the Sobel and Scharr edge detector algorithms as a mean to isolate the net structure. Moreover, a damage detection algorithm based on region growth of background pixels (binary zeros) using 4-connectivity, was developed. Sletta concluded that, among the algorithms tested and developed, edge detection was the better segmentation method; it managed to separate the net wall from the background under fairly ideal conditions - independently of sea and net color. However, the success of the algorithm was heavily varying with; reduced image quality, as resulting from camera movement or lens focusing issues; varying light conditions, such as light from solar flares and shadows; as well as foreign objects obstructing the net structure. The region growth detection algorithm performed well, but suffered from inaccurate segmentation results (Sletta, 2013).

1.3 Scope of this Study

Previous research on damage detection have been limited to particular test settings and partially idealized scenarios. In this thesis, the focus will be on developing a robust hole detection algorithm that is applicable to the challenges encountered during normal ROV operations.

The developed workflows and algorithms will therefore be evaluated based on their performance when applied to video footage from a sea cage ROV operation captured at Kattholmen salmon aquaculture farm in Norway, as provided by SINTEF F&A. Furthermore, since this footage contains no visible net damage, the video material collected by (Sletta, 2013) and (Olsen, 2013) will be used to address the performance of the actual hole detection methods. The two video sources differ in several aspects, as discussed later in Section 1.5, and will therefore enforce adaptability in the developed system.

The essence of the damage assessment approach that will be developed in this thesis can be formulated as follows:

> At any point where the *background* is obstructed from view either by the net, fish, ropes, growth or any other foreign objects, damage can either not be assessed, or the net structure is intact. However, if there is a continuous, large area of background present somewhere in the view, then this area indicates the lack of net structure and most likely net damage.

In other words, we cannot detect net damage if we cannot see the net visually, and the only scenario in which there is an area in the view that leads directly to the background, is if there is a lack of net structure in that area. Assessing the structural integrity of the net therefore becomes a task of isolating the background from *every* foreground (FG) object, including the net itself, and detecting larger, continuous areas where no foreground objects are present.

The methodology of isolating, or segmenting, the background from the foreground structures bears close similarity to the net tracking approaches used by (Jakobsen, 2011), (Sletta, 2013) and (Olsen, 2013). An important distinction, however, is that these studies only tried to find the net structure, whereas the methodology of this thesis implies that all objects are found - including foreign objects, fish and algae growth. Since previous works showed that finding the net structure without occluding foreign elements is likely doable, the remaining task therefore becomes to isolate every foreign object from the background as well.

In order to segment the background from all foreground objects, a combination of background segmentation approaches - each able to isolate a certain kind of objects under particular conditions - will be incorporated into one single background segmentation system. The final system will combine optical flow, edge detection and temporal background segmentation. Moreover, each method will calculate and combine segmentations from multiple image channels, in total providing several layers of redundancy. The system will aim to best assure robustness towards the challenges that has not yet been encountered in the video material analyzed, as well as to manage several cases analyzed that will be described later

All of the segmentation methods require some measure of binarization. Automatic binarization would be the preferred option, but was concluded unsuitable for net inspection by (Sletta, 2013). The assumption that suboptimal, but robust, static binarization parameters consistently will highlight the characteristic points from each image channels will be made, a property that the final combinatorial background segmentation system will utilize in order to produce a comprehensive combined segmentation of the background, despite operating on static parameters.

Temporal background segmentation can, in theory, isolate any object, regardless of its shape, orientation, size, movement, texture or color, from a model based background estimate, as long as it differentiates itself from this estimate in some fashion. However, reliably calculating an accurate background estimate is no trivial task in view of all the challenges encountered during underwater net inspection. Therefore, optical flow analysis and edge detection will be used in order to aid the temporal background segmentation algorithm, while also providing their unaccompanied contribution.

Furthermore, several measures will be made in order to fit a conventional temporal background segmentation method to what may be considered a rather unorthodox application for this kind of procedure. Specifically, typical applications for temporal background segmentation implies a stationary camera, whereas an ROV-mounted camera is in constant motion. By assuming that the background indeed has zero motion relative to the camera, while all foreground objects are always in relative motion, the theoretical basis of the temporal background segmentation still holds. However, if the ROV then were to stop, this assumption would no longer be true, making all foreground objects qualify as background, eventually giving erroneous segmentation results. By feeding the current detected foreground - combined from a edge detector, motion estimate, and the previously detected temporal background segmentation result - back to the temporal background segmentation algorithm, this scenario will be attempted handled.

In terms of assessing net damage, the segmentation result will be analyzed using a combination of filtering and morphological operations where the size of the structuring element can be adjusted to only detect damages above some user-defined size. Contrary to detecting damage with region growing as purposed by (Sletta, 2013), this approach will not require a perfectly continuous net structure.

Although theoretically possible with the methods utilized, detecting single mask damages will not be the main focus of the system developed in this thesis. Instead, robustness towards changing scene conditions and yet to be encountered challenges will be the main priority.

Real-time capability of the system designed in this thesis is not an absolute requirement for the tethered ROV-platform it would be intended for, but due to limitations in the hardware setup used for development, only real-time capable algorithms will be considered.

In the next two sections, a detailed overview of the challenges found from analyzing the video material by SINTEF F&A and (Sletta, 2013) and (Olsen, 2013) will be introduced, their relevance to this thesis discussed, and a general problem description given. In Section 1.6, the structure of the subsequent chapters of this thesis will reviewed.

1.4 Sea Cage Structures

In this section, a brief overview of common aquaculture sea cage net structures and damage types will be introduced. The images are snapshots from the video footage by SINTEF F&A and (Sletta, 2013) and (Olsen, 2013) that has been converted to grayscale and had their contrast greatly amplified for illustrative purposes.

In Figure 1.1, a simplified schematic of an aquaculture cage is shown. The three images in Figure 1.1 show three net structures that can be commonly viewed from inside a cage. A clearer view of single and double net structures without algae growth, along with an image displaying several ropes and foreground fish, can be viewed in Figure 1.2. Sea cages often have a volume ranging between 20 000 and 80 000 m^3 (Jensen et al., 2010).

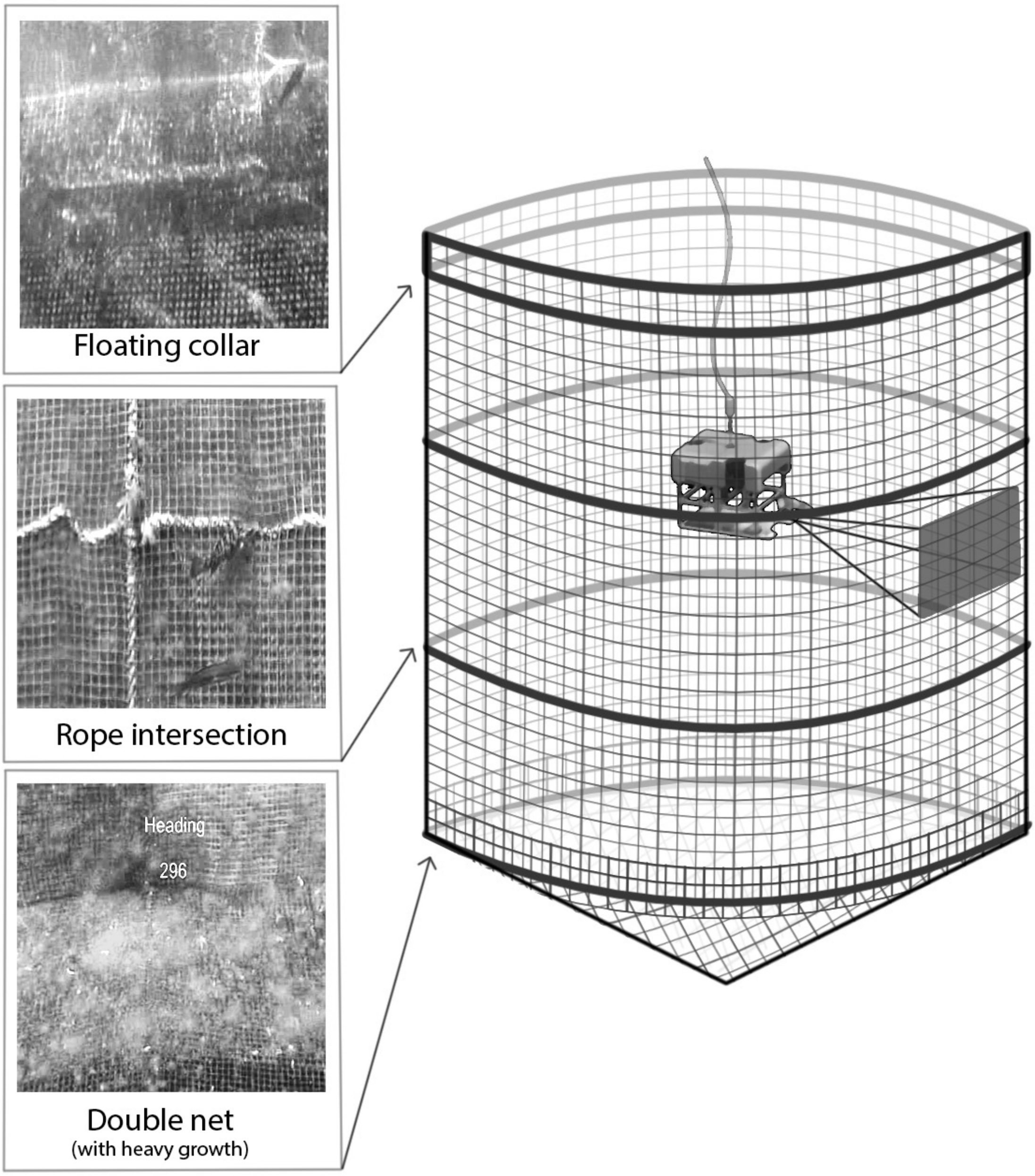
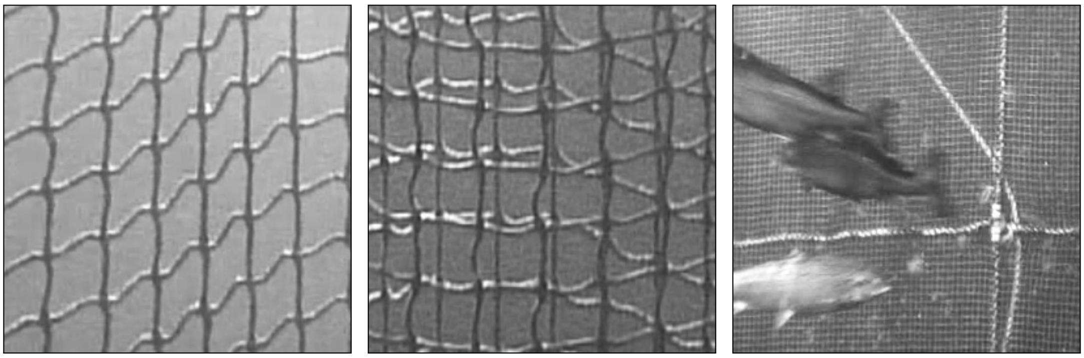


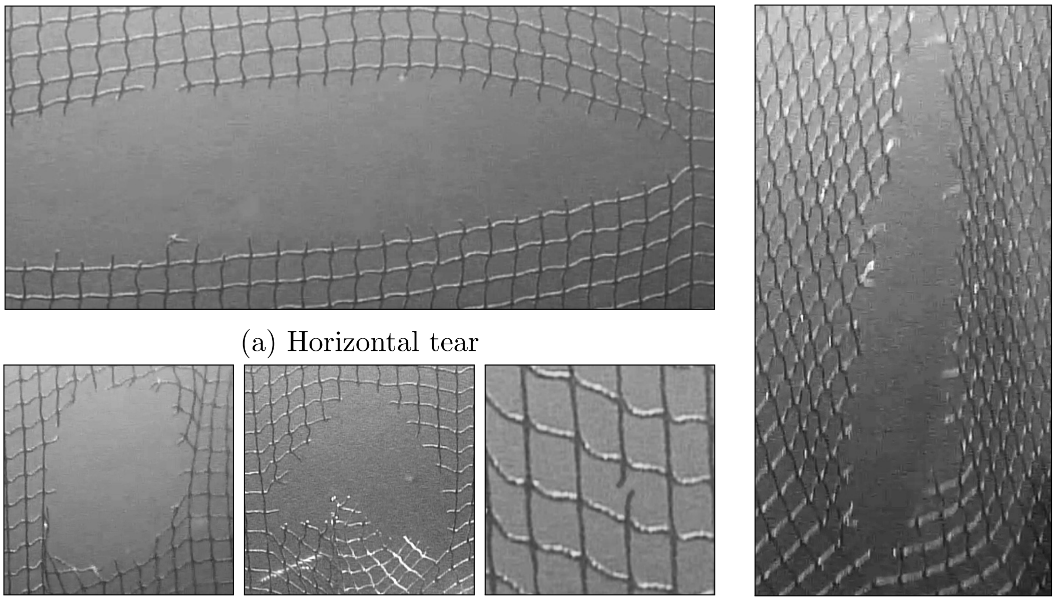
Figure 1.1: A simplified illustration of a tethered ROV inspecting a cage with camera vision



(a) Single net (b) Double net (c) Ropes and fish

Figure 1.2: Common net structures, ropes and foreground fish inside a sea cage

A number of potential net damage types are illustrated in Figure 1.3. According to Heide and Moe, the L-tear is the net tear type that is most easily escaped, as it leaves a clear, breachable hole for the fish to pass through in most cases. On the other hand, a vertical tear typically leaves a smaller gap for the fish to escape through, and is considered less critical. Tear damages can appear as single masks, or stretch up to several meters (Heide & Moe, 2004).



(a) Horizontal tear

(b) Hole (c) L-tear (d) Single mask (e) Vertical tear

Figure 1.3: Illustrative net damage types as they might occur in a real sea cage. Image (d) was manipulated to show a single mask tear, as no such damage was found in the real footage

1.5 Challenges of Automatic Damage Assessment

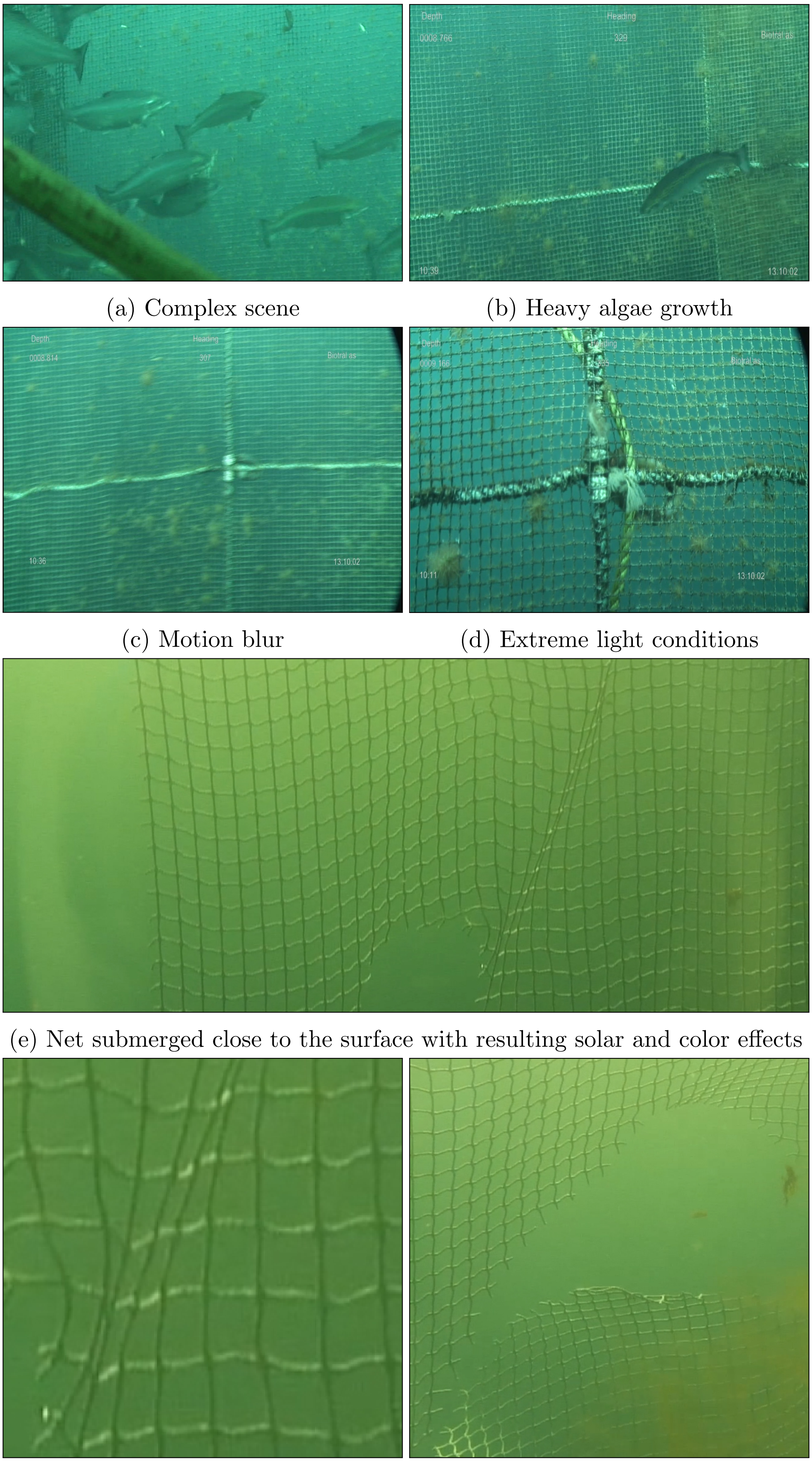
The image recognition capabilities of the human eye still heavily outperforms todays computer vision algorithms for most problem situations, and detecting inconsistency in a moving texture, such as a net, poses a whole set of challenges for a computer vision

1. Varying color, contrast, lightning, scale, rotation and perspective
2. View obstruction from fish, ropes, algae growth and foreign objects
3. Multiple net structures
4. Motion blur, focus blur and limited camera resolution
5. Everything moves, no static elements - usually
6. Limited reproduction of details in single image channels and under extreme light conditions
7. Discretization noise
8. Capturing errors in the simulated video material

Table 1.1: Overview of challenges associated with net damage assessment

algorithm that most human observers would not even notice. Despite its difficulties, the future potential of an automatic damage assessment camera solution has ranked it as a meaningful research topic by (Taylor & Kelly, 2010). In this section, several example images from captured video footage will be presented, highlighting some of the common challenges and uncertainties related to assessing net integrity with computer vision, and difficulties concerning underwater video capture in general. The challenges found will then be evaluated in terms of the their relevance in this thesis.

The work of this thesis is based on two sources of video material: video capture from a real ROV-inspection collected at Kattholmen in Norway by SINTEF F&A, and video samples that artificially simulates various net damage types using a motorized setup collected by (Sletta, 2013) and (Olsen, 2013) - for simplicity referred to as S&O. The images (a)-(d) in Figure 1.4 are extracted from the video material provided by SINTEF, while the images (e)-(g) are snapshots from the video material by S&O. The images presented have been cropped, but are otherwise in their original state. A summary of all the challenges discussed in this section can be found in Table 1.1.



(a) Complex scene

(b) Heavy algae growth

(c) Motion blur

(d) Extreme light conditions

(e) Net submerged close to the surface with resulting solar and color effects

(f) Top-lit net

(g) Shadow areas from L-tear

Figure 1.4: Common scenes from an ROV net inspection

SINTEF F&A Video Evaluation

The view in Figure 1.4a could be from an ROV that is changing attitude or overall position in the net. The scene is complex, with the tether cable and fish obstructing the view of the net, while the net itself is slightly out of focus with patches of shadow as well as algae growth. Moreover, the detail rendering is restrained at this distance by a limited camera resolution, which makes seeing individual net threads difficult. Due to the limited visibility, scenes such as this will not be evaluated by the final system. The foreground elements, however, are still very much relevant.

In Figure 1.4b, the ROV has been positioned in front of the net wall, giving an overall better visibility of the net structure. Particularly visible in this view is a large area of algae growth that is fully or partially covering areas of the net. The fully covered net areas are normally unfit for visual inspection, while the partially covered areas are usually well examinable by a human operator. With the methodology of this thesis, on the other hand, the algae growth overlaying the net differs from the background in terms of brightness and color, and will therefore be classified as foreground. This effectively means that net damaged covered in algae growth will be undetected.

A scene similar to Figure 1.4b is presented in Figure 1.4c. The entire image is heavily blurred by the relative motion between the net and the ROV. Severe motion blur will often erase detailed pixel information in the direction of motion, making the frame infeasible for accurate visual inspection. Since motion blur regularly is seen in the analyzed video material, the damage assessment scheme in this thesis will focus on detecting larger damages, since the occasional lack of image content otherwise would make single mask damages appear erroneously at a high rate.

The scene in Figure 1.4d has a lucrative perspective and net distance, with a sharp net structure due to a slowly moving camera. On the other hand, it clearly shows the extreme variations in light intensity affecting the net structure, and the consistent spread of algae growth often present during visual net inspection. Also, the upper right part of the image is clearly brighter than the bottom left, with a gradient transition in between. The smoothness of the algae growth will make it difficult to detect for an edge detector algorithm, and the response from optical flow algorithms will be limited in this scene, since there is very little motion present. The temporal background segmentation should in theory be able to segment all structures, given that the structures stand out from the background in some manner. This assumption can, however, not always be made, as discussed later.

Common for the video material by SINTEF F&A, is a background scene that is slowly changing, with few sporadic alterations in light conditions apart from shadows in the net structure itself. The background scene is also largely smooth and homogeneous, and contains no distinctive colors, textures or objects. Foreground elements seem to occasionally have little motion relative to the camera - such as when the ROV is changing direction, making it momentarily stand still - but are usually in consistent motion. These observations seem estimable for some temporal background segmentation model, and generally applicable for optical flow analysis and edge detection.

S&O Video Evaluation

In the video material by S&O, as illustrated in Figure 1.4 (e)-(g), the solar and color compositional effects from setting up a net close to the water surface are visible. In

Figure 1.4e, the sun generated a rapidly flickering flare in the camera's optical system visible as bright a halo in the video feed. The video feed also revealed sun spots that moved across the scene, momentarily brightening entire image regions, while the background itself displayed small illumination changes at a frequent rate. Moreover, the magnified view in Figure 1.4f clearly displays how each thread of the net were top-lit, giving each horizontal thread both a bright and a dark appearance.

The video samples by S&O include scenes with stationary net structure and artificial growth, as well as scenes with motorized, vertical motion. In these videos, an assumption of a foreground in continuous motion will therefore not hold.

In his thesis, (Olsen, 2013) discovered that the video material he produced with (Sletta, 2013) suffered from an irregular frame flow due to RAM issues with in capturing equipment. This irregularity causes a skipping effect in the video stream, and appears to be successive frame duplicates. This might be a concern with regards to evaluation of optical flow.

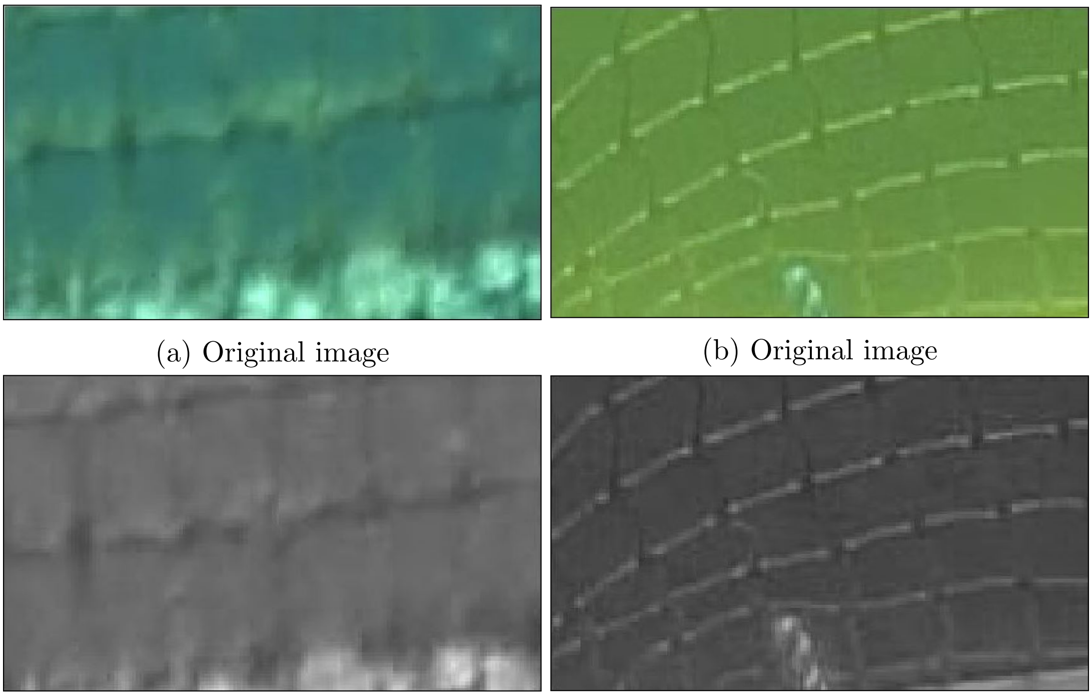
Shared Quality Issues

For both video materials, reduced image quality for certain scenarios was observed.

Although all foreground structures seem to clearly distinct themselves from the background in Figure 1.4d, each individual image channel does not necessarily have equally good object separation for each type of foreground element. In Figure 1.5, a close-up image of Figure 1.4d, and a similar close-up image from the material by S&O, is displayed. It is clearly visible in Figure 1.5 (a) and (c) that foreground structure that is well defined in the original image indeed can be poorly defined in a single image channel. Combining information from multiple image channels will be an attempt to control this limitation.

Furthermore, in Figure 1.5 (b) and (d), the diminishing effect on image detail an extreme light settings can give is shown, where neither the original image nor the single color channel reproduces the net structure well. In this case, the net cannot be distinguished from the background, and unless the damage assessment algorithm is designed to ignore minor net damages, this scenario will yield a false positive detection result.

In Figure 1.5 (b) and (d), what appears to be discretization noise markedly reduces the image quality of the smooth image regions. Color noise and so-called *salt* and *pepper* noise, on the other hand, seems minimal.



(a) Original image

(b) Original image

(c) Blue color channel

(d) Blue color channel

Figure 1.5: Video material quality limitations

Comparison and Conclusion

The simulated net videos by S&O incorporate multiple challenges that most likely would not apply to net inspection at increased depths, as evident from comparing this material to the real video footage captured by SINTEF F&A. The rapid changes in light conditions, background instability, solar flares and sunspots in S&O's video material, makes these videos radically different in several aspects, and if these challenges were to be managed, different algorithmic choices would have to be made than what would be required by the material from SINTEF F&A.

The main focus will be to best handle the video material by SINTEF F&A, since this represents a realistic problem scenario. Algorithmic choices from this decision will therefore potentially result in a suboptimal response when the damage assessment algorithm is tested on the material by S&O. On the other hand, these tests will also truly trial the robustness of the developed system, as it will be evaluated based on challenges it is not designed to handle.

1.6 Structure of this Thesis

This thesis has been organized in the following manner:

Chapter 2, Literature Review and Background Material: In this chapter, previous research that in some manner influence the decisions and reasoning made in the subsequent chapters will be presented. Also, all methods adopted from other literature will be covered in detail here. Section 2.9 might be of particular interest to the reader, as it summarizes the reviewed literature in the context of net inspection, and gives an overview of the methods proposed in the next chapter.

Chapter 3, Methods and Materials: This chapter will cover all the suggested solutions and algorithms that ultimately will lead to the design of the final damage assessment system of this thesis. Several of the procedures presented in this chapter will later be used solely for experimentation and comparison as a part of the iterative development that has lead to the final system. Since multiple of these methods were designed or included based on initial problem analyses, some early experiments will be documented in this chapter as well. In addition, the materials and tools that will be used for experimentation and performance evaluation will be introduced. The chapter is organized in a bottom-up fashion, where the most basic components are described first, while the combinatorial design and final system are described at the end of the chapter, in Section 3.9 and 3.11, respectively.

Chapter 4, Experimental Results: Components: In this chapter, experiments documenting internal component behavior, parameter selections and the design decisions that has lead to the development of the final system design will be analyzed, discussed and concluded upon. In general, this chapter contains most of the fundamental research conducted through the work of this thesis. The content of this chapter has been give a modular structure, where the section of each individual component test include its own discussion and conclusion. As such, the reader might find this chapter useful as a reference, since each component test can be read without the full knowledge of previous experiments. Sections of particular interest are Section 4.3, 4.4 and 4.6, where the initial verdict of edge detection, optical flow and temporal background segmentation - the three main components of the final system - is given.

Chapter 5, Experimental Results: Final System: This chapter is dedicated to displaying the performance and analysis material for the final system tests, results of which will be discussed and concluded upon in Chapter 6 and 7, respectively. Although preferred, the reader does not need to have knowledge of the experimental results from individual components in Chapter 4 before reading this chapter.

Chapter 6, Discussion: Based on the results from Chapter 5, the applicability of the total net damage assessment system developed in this thesis will in this chapter be evaluated. While some of the most important observations from Chapter 4 will be revised in this chapter as well, the main focus will be to evaluate the combinatorial and damage detection methodologies purposed throughout this thesis. Furthermore, benefits and shortcomings of the final system will here be presented, and its prospect to practical implementations given. Lastly, suggestions for further research will made.

Chapter 7, Conclusion: A brief summary of the purpose and most important findings of this thesis will here be reported, and a final word given.

Appendix: Throughout this thesis, a large amount of data, illustrations, graphs, and images will be introduced and discussed. While the majority of these graphics will be presented in their respective chapters, several larger collections of graphics will be contained within this chapter, both for spatial reasons, as well as to allow for easier comparison. Furthermore, a summary of the terminology used will be located here.

Chapter 2

Literature Review and Background Material

In this chapter, the background literature supporting the system design decisions conducted in Chapter 3 is presented. The literature reviewed gives a detailed look at several research publications of central methods used in this report, as well as some fundamental operations and commonly used computer vision techniques that will be assumed known by the reader in subsequent chapters.

The structure of this chapter follows the workflow outlined in Figure 2.1, which loosely resembles a common path of action in systems incorporating background segmentation - as will be seen at several occasions throughout this chapter. Finally, in Section 2.9, the reviewed literature will be evaluated with regards to the challenges encountered during underwater net inspection, and possible ways to overcome these challenges introduced.

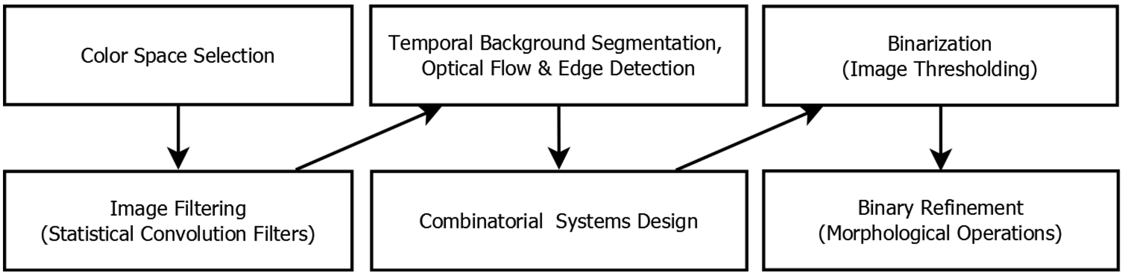
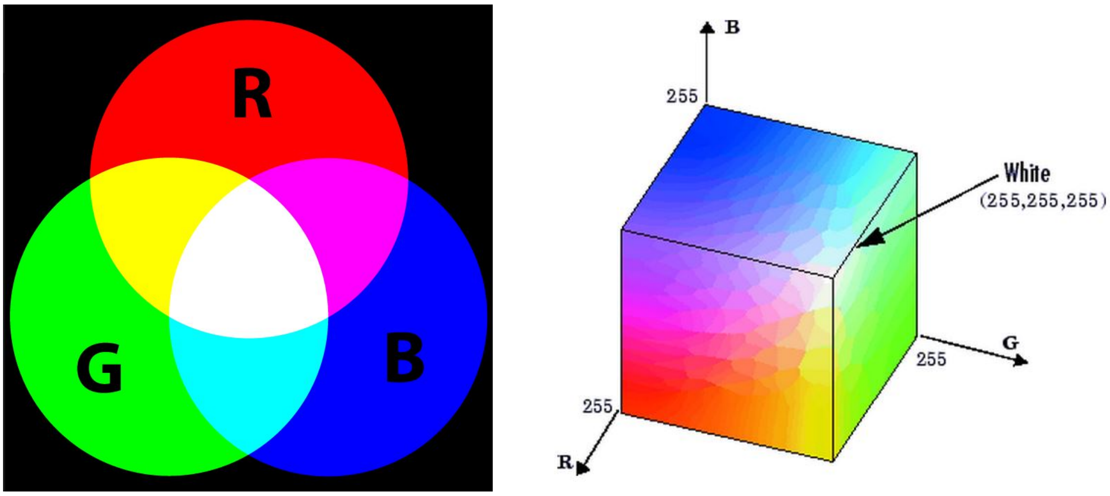

Color Space Selection
Temporal Background Segmentation, Optical Flow & Edge Detection
Binarization (Image Thresholding)
Image Filtering (Statistical Convolution Filters)
Combinatorial Systems Design
Binary Refinement (Morphological Operations)

Figure 2.1: Workflow outline of Chapter 2

2.1 Color Spaces

Images are typically represented digitally by large 2D-matrices where each element, or *pixel*, consists of a numeric tuple of three or four values. These numeric tuples are designed to accurately mirror the normal human perception of color (Joblove & Greenberg, 1978). Each numeric value, or *channel*, contains a particular class of information about the image, such as the content of a primary color - red, green or blue - or properties such as luminance (brightness) and chrominance (hue and saturation) (Plataniotis & Venetsanopoulos, 2000). A set of image channel properties, and their numeric representation, is normally referred to as a *color model*, whereas a *color space*



(a) RGB is an additive color space

(b) RGB 3D color space representation

Figure 2.2: RGB color space properties. Retrieved March 25, 2014 from: (left) https://bpiinc.wordpress.com/tag/rgb/ and (right) http://www.mathworks.se /help/image

defines the colors that can be produced by mixing the information contained in the image channels in a particular manner.

2.1.1 Red Green Blue

The Red Green and Blue (RGB) color space is composed of the three primary colors that combined can produce the greatest number of displayable colors, namely red, green and blue (Joblove & Greenberg, 1978). Most digital monitors and sensory equipment produce their colors by combining the same primaries, which makes the RGB color space practical for many computational purposes (Plataniotis & Venetsanopoulos, 2000). Colors are produced by mixing the three primary components in an additive manner, where adding more color increases the brightness of the result, as illustrated in Figure 2.2a. A 3D model of the RGB color space is illustrated in Figure 2.2b. In digital processing, each color channel is typically represented numerically by integers values ranging between 0 and 255.

A well known disadvantage of the RGB color space, is its inability to uniquely describe a color without a given white point reference and a gamma correction value (Plataniotis & Venetsanopoulos, 2000). Moreover, for instance, a single red value can be used to describe a variety of colors depending on the values of the green and blue pixel components. For this reason, the RGB color space is deprecated for many computer vision purposes where color is a decisive element (Plataniotis & Venetsanopoulos, 2000).

2.1.2 The Hue Saturation and Intensity Family

The Hue Saturation and Intensity (HSI) family of color spaces was introduced by Joblove and Greenberg in 1978 as a mean to better describe unique colors digitally in a manner that correlates well with the human perception of color (Joblove & Green-

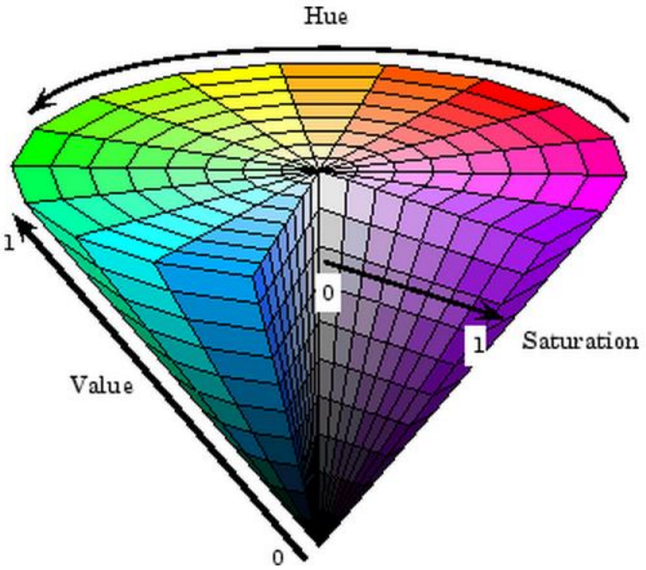


Figure 2.3: HSV color space 3D representation. Retrieved March 25, 2014 from http://www.mathworks.se/help/images

berg, 1978). The HSI family of spaces consists of the Hue Saturation and Value (HSV) color space, the Hue Lightness and Saturation (HLS) color space, as well as HSI itself. These are all slightly different representations the RGB color space transformed into cylindrical coordinates, and therefore possess similar qualities (Plataniotis & Venetsanopoulos, 2000).

A major benefit of the HSI family of color spaces is their separation of chromatic values from luminance. In practical terms, this mirrors the human ability to uniquely identify a particular color despite varying light conditions, atmospheric scattering or even through haze (Joblove & Greenberg, 1978)(Plataniotis & Venetsanopoulos, 2000). The color tone is stored in the hue channel, while the pureness of the tone (the infusion of white) is given by the saturation channel, and the last parameter, given by either value, lightness or intensity, describes the overall brightness of the color. This relation can be viewed in Figure 2.3, which displays the HSV 3D color space for a normalized value range. The ability to separate chroma from luminance, makes the HSI family of color spaces favorable over the RGB color space in applications where color features are used as decisive elements, such as in image segmentation (Plataniotis & Venetsanopoulos, 2000).

However, the HSI family also has a significant drawback: Along the center axis of the HSI color spaces is a singularity for the hue component, which means that any tone of gray will be undefined in the hue channel. Therefore, any practical implementation of the hue channel will need to handle this scenario.

2.2 Statistical Convolution Filters

Throughout this thesis, classical image filters such as the *median*, *averaging* and *Gaussian* filters, will be utilized for noise reduction, to induce blur, extract features or merely for comparison purposes. These filters are related in that they calculate the pixel value for all image points according to the neighborhood of each individual pixel through *template convolution*. The main points of these filter operators are given here; for an in-depth explanation, the reader is referred to (Nixon & Aguado, 2002), from where the content of this section has been collected.

The average filter operator is characterized by a evenly weighted convolution template, where the sum of the weights equals unity. With an averaging filter, high-contrast points will get suppressed, making low-frequency information more visible. A 3x3 averaging template can be viewed in Figure 2.4a.

The Gaussian filter operator is in literature considered optimal for image smoothing. In addition, image noise is often approximated to a Gaussian distribution, in which case the Gaussian filter operator is much suited for noise reduction. The template weights take the shape of a 3D Gaussian distribution controlled by the variance σ^2 - as illustrated for a 3x3 template and a generic 3D representation in Figure 2.4b and 2.4c, respectively.

The median filter operator sets the median value of all pixels covered by the convolution template as the resulting value; this way, outliers in the evaluated region are efficiently suppressed, which makes the median operator ideal for removing so-called *salt and pepper* noise, while preserving detail.

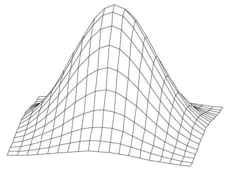
As can be seen from Figure 2.5, the averaging filter has a slightly stronger low-pass filtering effect than the Gaussian filter, while the median filter removes more noise while retaining image detail (Nixon & Aguado, 2002).

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

(a) Average 3x3

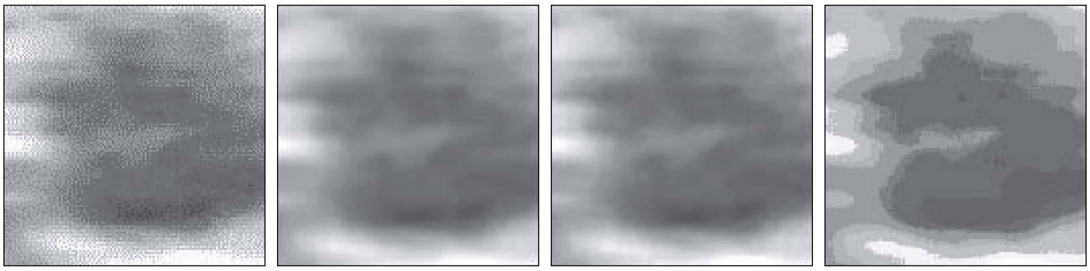
0.0929	0.1190	0.0929
0.1190	0.1525	0.1190
0.0929	0.1190	0.0929

(b) Gaussian 3x3, $\sigma = \sqrt{2}$



(c) Gaussian 3D representation

Figure 2.4: Filter operator template examples (Nixon & Aguado, 2002)



(a) Original (b) Average (c) Gaussian (d) Median

Figure 2.5: Comparison of statistical filter operators (Nixon & Aguado, 2002)

2.3 Temporal Background Segmentation

Temporal background segmentation is one of the basic, low-level operations that is often represented in a typical video surveillance workflow, with the intention of separating an expected scene (a background) from unexpected entities (foreground elements) (Cristani, Farenzena, Bloisi, & Murino, 2010). Traditional applications for temporal background segmentation are typically related to tracking, detecting and recognizing people or vehicles with stationary cameras mounted alongside roads and in urban environments. A background segmentation procedure can normally be divided into two main tasks:

1. Initialization of some appropriate *background model* from acquired data or knowledge about the scene
2. Maintenance and *update* of the background model to account for permanent scene changes or recurring background elements

There are numerous approaches to solving the two main tasks of temporal background segmentation. An excellent, extensive review of today's most used designs and their limitations can be found in (Cristani et al., 2010), while a compact revision of some of the fundamental methods was made by (Piccardi, 2004). Most modern literature divide temporal background segmentation techniques into *per-pixel*, *per-region* and *per-frame* based methods. Cristani further divides these methods into *mono-* and *multi*modal, as well as *non-parametric* categories.

In per-pixel based methods, each pixel is considered an individual process, whereas region and frame based methods evaluates regions of pixels and entire frames, respectively, at a higher level. The assumption of pixel independence is considered a significant drawback of the per-pixel algorithms. However, they balance segmentation accuracy and speed in a desirable manner, and are often used in real-time applications. Per-region methods will in some cases provide better accuracy than per-pixel algorithms by considering interpixel relations, but at the cost of higher computational requirements, making them less suited for real-time usage. Per-frame algorithms gives the benefit of analyzing an entire frame as a whole, allowing for complex scene analysis. However, most of today's approaches require offline training on some collected data set. Collecting this data can in itself be an issue, but also raises question as to how the background model should be updated. Classification, on the other hand, is usually suited for real-time usage (Cristani et al., 2010).

Multimodal methods are characterized by their ability to represent backgrounds with several appearances, for instance with swaying trees or waves. Algorithms of this class based on Gaussian Mixture Models (GMM) have proven successful for many applications, and several improvements have been made over its original statement, as summarized by (Bouwmans, El Baf, Vachon, et al., 2008). Monomodal techniques, on the other hand, model the background as if it has a single appearance - that is, without constantly recurring elements. Common monomodal methods are: Running Gaussian Average, as well as Temporal Median, Mode and Averaging filters. Kernel Density Estimation (KDE) is a popular technique for non-parametric multimodal background modeling, often used in situations where selecting model parameters is difficult (Cristani et al., 2010)(Piccardi, 2004).

Although runtime performance will not be the main focus of this thesis, the final system produced could potentially benefit of having real-time capabilities in the future - in particular if an untethered AUV were to be developed for the purpose. Additionally, as already discussed in detail in Section 1.5, the background appears homogeneous and well behaved - likely suited for a monomodal representation. Therefore, research on per-pixel monomodal background estimation algorithms will now be revisited. Per-frame algorithms will not be covered due to the lack of training material.

2.3.1 Temporal Motion Filtering

Several temporal buffer based background segmentation approaches have been studied by R. Cucchiara, M. Piccardi and their team of researchers, while a generic per-pixel based monomodal implementation of a temporal median filter was suggested for background subtraction by (Lo & Velastin, 2001). Due to their direct relevance to the system designed later in this thesis, their works will now be thoroughly reviewed.

In Lo's implementation of a temporal median background removal system, the background model consisted of a buffer structure that stored the n last frames from a video sequence. Each pixel in the current background was updated by calculating the median value from that pixel's (x,y)-position in the previously buffered frames, as illustrated in Figure 2.6. The current background was then subtracted from the current video frame, and converted to a binary format with some unspecified thresholding technique. Lo's background removal workflow has been re-illustrated in Figure 2.7 (Lo & Velastin, 2001).

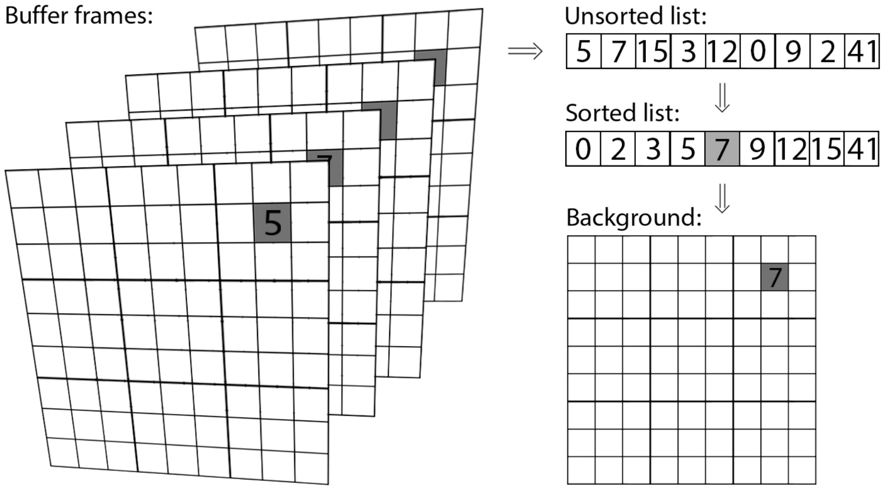


Figure 2.6: A basic temporal median background update procedure

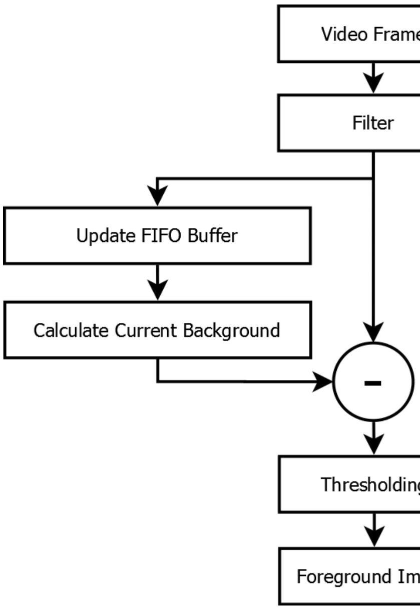


Figure 2.7: Background removal for a single frame with Lo's temporal median background model

In (Cucchiara & Piccardi, 1999), two different sets of algorithms were incorporated to handle day and nighttime motion-based vehicle tracking used in a Real-time Traffic Surveillance System (RTSS). The system was split into a low-level and high-level part: the low-level part detected moving points in daylight using a *spatio-temporal* segmentation called *double-differencing* (Kameda & Minoh, 1996) on three consecutive frames, followed by a morphological closing operation and region growing for object classification; the higher-level system validated the segmentation results using a forward chaining (Hayes-Roth, Waterman, & Lenat, 1983) tracking system. The extra validation steps ensured robust tracking despite changing scene conditions.

In (Cucchiara, Grana, Piccardi, & Prati, 2000), a background estimation approach based on the *mean*, *median* and *mode* of previously buffered frames was compared in terms of long term segmentation performance of an *a priori* unknown background using a Statistical & Knowledge Based (S&KB) background update (see Section 2.3.2). It was concluded in their study that the mode operator modeled the background better, while the mean was worse - in particular for high contrast images and few buffered frames. However, in order to satisfy real-time constrains in the selection of buffer size and sampling interval, their final solution incorporated the median operator, which better approximated the background while upholding their desired learning rate.

2.3.2 Background Update Schemes

The two main approaches to updating a background model has by (Elgammal, Harwood, & Davis, 2000) been categorized as:

1. Selective background update

2. Blind background update

In a *selective update* scheme, a each individual pixel in a new frame is only added to the background model if it in the previous background segmentation was classified as a background point. The benefit of this approach, is that pixels classified as foreground avoid interfering with the background model, and additionally allows for saved computations, as their computation would have been excessive (Koller, Weber, & Malik, 1993). However, this could also lead to situations where permanent background changes - momentarily classified as foreground - would forever be excluded from further model updates - usually referred to as a *deadlock* situation (Cucchiara et al., 2000)(Elgammal et al., 2000).

A *blind update* scheme, on the other hand, embeds every pixel in each new frame into the existing background model. This approach will never encounter a deadlock situation, but will also allow foreground pixels to erroneously influence the background model, reducing detection accuracy. Increasing the time window over which samples are collected - the buffer size n in temporal filters - would reduce this error, but also limit the *learning rate* - the background model's ability to adapt to continuous scene changes (Elgammal et al., 2000).

Multiple authors have purposed ways to overcome these issues, as well as manners of handling application specific problems of similar nature. An interesting approach suggested by (Cucchiara et al., 2000), the S&KB update method, utilizes motion analysis on previous frames in order to reap the benefits from a selective update scheme, while avoiding deadlock by categorizing stationary foreground elements as background.

Statistical & Knowledge Based Background Update

The statistically based update method suggested in (Cucchiara et al., 2000) was initially formulated as follows:

$$B_t = U(I_t, I_{t-\triangle t}, \, ... \, , \, I_{t-(n-1)\triangle t} \, , \, w_b B_{t-\triangle t} \,) \tag{2.1}$$

where B_t and I_t are the approximated background and image frame at time t, respectively, n is the buffer size of previous frames organized in a First In First Out (FIFO) manner, $\triangle t$ is the sampling interval and w_b is the preservation weight of the latest background estimate. Moreover, the function U represents the statistical operation - mean, median or mode - performed. The exact meaning of the term $w_b B_{t-\triangle t}$ in Equation 2.1 for a practical implementation was not specified in (Cucchiara et al., 2000), but the w_b parameter was later interpreted by (Piccardi, 2004) as the number of times the background frame $B_{t-\triangle t}$ were to be appended to the frame buffer during statistical calculations. The optimal parameter values were found to be $n = 9$, $w_b = 2$ and $\triangle t = 10T_f - 50T_f$.

The optimal selection of the sampling interval $\triangle t$ varied during each video sequence: a short sampling interval would give a responsive classification and background update, but would easily include slowly moving objects into the background model; a long sampling interval, on the other hand, would give an unresponsive algorithm with a poor learning rate - neither of which were desired. Because of this, a modification of the selective background update scheme was suggested, where points classified as foreground pixels by the segmentation would only be excluded from the background update if their optical flow was non-zero. This way, the deadlock issue associated with a selective update scheme would be avoided, while allowing real-time performance due to a limited buffer size n, as well as a short sampling interval $\triangle t$ giving responsive classification and an adequate learning rate. Mathematically, this improved background update, namely the S&KB update method, was formulated as follows:

$$B_t = \begin{cases} B_{t-\triangle t}, & \text{if } I_t \in \{FG_t^k \text{ and } OF^k > TH\} \\ U(I_t, I_{t-\triangle t}, \, \dots \, , \, I_{t-(n-1)\triangle t} \, , \, w_b B_{t-\triangle t} \,), & \text{otherwise} \end{cases} \tag{2.2}$$

where OF^k is the calculated optical flow for some foreground point FG^k, and TH is some predefined threshold value.

2.3.3 Comparison of Methodologies

The temporal background segmentation approaches studied above were designed for specific applications with different concerns and goals. In general, a few observations can be made:

The temporal median background model implemented in (Lo & Velastin, 2001) utilized a blind update scheme, as discussed in Section 2.3.2. In video streams with a high presence of slowly moving foreground objects, these objects are likely to influence the background estimate unless a large buffer is used - at increased computational costs and a lower responsiveness to background changes. In (Piccardi, 2004), this generic temporal median background model is criticized for its high memory requirements, lack of a rigorous statistical model, and its absence of a deviation measure for using adaptive parameters.

The modified selective update scheme, S&KB, suggested by (Cucchiara et al., 2000) addresses several of the hypothetical downsides with the background model implemented by (Lo & Velastin, 2001). However, by doing so, additional complexity with regards to selecting, implementing and tuning an optical flow algorithm emerged - including the extra tuning parameters embedded in Equation 2.2. Furthermore, the S&KB update scheme was intended to include foreground objects that became stationary in the background estimate. In the application of this thesis, this feature will erroneously allow the foreground to influence the background estimate if the ROV has no motion relative to the net.

A general concern regarding buffer based background models, is the manner of which they are initialized - addressed by neither (Cucchiara et al., 2000) or (Lo & Velastin, 2001). Before the buffer is completely full, the quality of the background estimate might be reduced: if the buffer is initialized in its full size with empty frames, the estimate might be invalid for an extended period of time depending on the statistical estimation operator used; with the median operator, at least half of the buffer must be filled before the frames with content appears in the background estimate.

For the S&KB background update scheme in particular, buffered frames will partially contain uninitialized pixels (usually set to black by default) if those pixels are continuously detected as being in motion during buffer initialization. In theory, if a pixels always is detected as in motion, it might never be initialized in the background estimate. Without particular handling, uninitialized pixels will normally appear as high-contrast points classified as foreground when used in combination with background subtraction.

2.4 Optical Flow Estimation

The positional displacement of moving pixels between two subsequent video frames is often referred to as their *optical flow* (OF), and can be thought of as the pixels' velocity with regards to time (Nixon & Aguado, 2002). Later on, optical flow will be used for background segmentation in combination with the S&KB update scheme, as well as a general purpose image indicator for motion.

The *optical flow constraint*, given in Equation 2.3, is a commonly used approximation when evaluating optical flow, where u and v are positional deviations for some pixel (x, y) over some time interval $\triangle t$ (Wedel & Cremers, 2011). This simplification ignores photometric differences between frames, such as temporal changes in illumination and shadows, or the effect of digital noise (Nixon & Aguado, 2002). In general, the accuracy of this motion estimate increases for smaller time intervals $\triangle t$.

$$I(x, y, t) = I(x + u,\ y + v,\ t + \triangle t) \tag{2.3}$$

Inside untextured regions and image objects, the motion of each particular pixel cannot be retrieved without additional information; similarly, the direction of motion can only be estimated in one dimension for the edge pixels of such regions. Solving this so-called *aperture problem* is at the base of many OF estimation algorithms (Wedel & Cremers, 2011).

2.4.1 Double Differencing for Motion Detection

For detecting image motion in general, (Kameda & Minoh, 1996) proposed the method of *double differencing*. Unlike many other optical flow algorithms, this approach does not find the precise movement of individual image pixels or pixel neighborhoods, it incorporates no flow constraints, nor handle the general aperture problem. It does, however, highlight image regions where motion is present - regardless of its origin.

The double differencing procedure suggested by (Kameda & Minoh, 1996) is illustrated in Figure 2.8. In Kameda's implementation, three subsequent frames were subtracted in pairs, generating so-called *difference images*. These subtracted images were then binarized individually, and combined using the binary AND operator. Binary ones in the resulting image were eventually combined into 4x4 square blocks, to rid of isolated noise pixels (Kameda & Minoh, 1996).

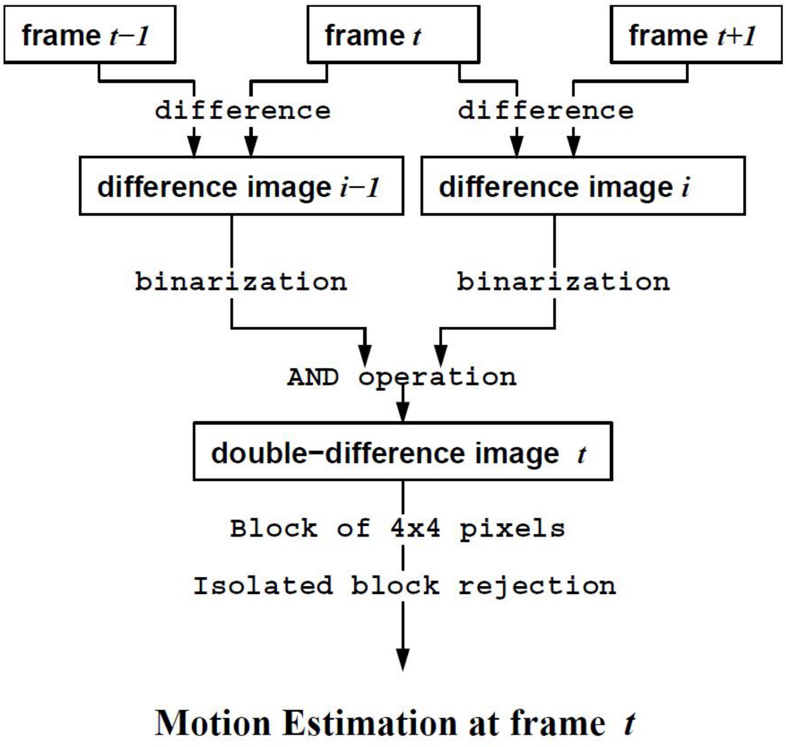


Figure 2.8: The double differencing motion detection procedure (Kameda & Minoh, 1996)

The double differencing motion detection method has the benefit of being fairly straight forward to implement, with low, deterministic computational requirements. However, it requires some measure of binarization and does not eliminate noise without pre- or post-processing - such as creating binary blocks.

2.5 Edge Detection and Blur Estimation

An *edge* can be described as an image point where a swift change in pixel intensity is present, such as along the boarder of two image objects (Nixon & Aguado, 2002). Edge detection algorithms are typically based on localizing contrast in pixel intensity by analyzing the first and second image derivatives. The popular Canny edge detector (Canny, 1986), aims to fulfill what may be considered the three main goals of most modern edge detector algorithms, which by (Nixon & Aguado, 2002) was summarized as:

1. Optimal *detection*: no falsely detected edges
2. Good *localization*: minimal distance between the true edge position and the detected edge
3. Single *response*: no multiple responses to a single image edge

A general property of edge detection algorithms, are their insensitivity to lightning changes, which makes them popular for image interpretation (Nixon & Aguado, 2002).

In this thesis, edge detection will be used for background segmentation in combination with the S&KB background update scheme, as well as a standalone feature detector. However, none of the popular edge detectors will be utilized; instead, a basic novel approach based on evaluating the sharpness of image points will be suggested.

An operation for determining local image sharpness was in (Sunkavalli, Joshi, Kang, Cohen, & Pfister, 2012) states as follows:

$$W_k = |I_k - G_\sigma \otimes I_k| \tag{2.4}$$

for some image I_k and a Gaussian smoothing filter G_σ with $\sigma = 3$. With this measure, a lower value W_k for some pixel (x,y) indicated blur, while a larger W_k indicated sharpness.

2.6 Combinatorial System Design

The final system of this thesis will utilize a combinatorial design, where the final segmentation result will be a junction of edge detection, optical flow and temporal background segmentation evaluated for multiple image channels. In 2005, (Karaman, Goldmann, Yu, & Sikora, 2005) reviewed the then state of the art methods for segmentation of static backgrounds, most of which employed combinations of various segmentation techniques on multiple image channels. Due to the direct relevance of Karaman's study to this thesis, some of the main concluding points will now be presented.

2.6.1 Combining Image Channels

All but one of the methods reviewed by (Karaman et al., 2005) somehow combined information from two or more image color channels in order to improve robustness of the result. Karaman concluded with the following:

> While color is a powerful clue for segmenting foreground objects from the background, grayscale information is simply not enough for robust detection. Its use should be limited to scenarios were color information is not available, such as night vision or infrared cameras (Karaman et al., 2005).

As such, combining image channel information is likely a valid approach for improving robustness also in the system of this thesis.

2.6.2 Combining Segmentation Methods

Regarding the methods reviewed that implemented combinatorial segmentation strategies, the following general conclusion was made by (Karaman et al., 2005):

> Edge information alone lacks robustness due to falsely detected edges but can improve the performance if used in combination with color. Generally the combination of complementary information (color, edge, motion) leads to higher performance (Karaman et al., 2005).

As will be seen later, this largely outline the design methodology of the system developed in this thesis.

2.7 Image Thresholding

Image thresholding can be described as the process of segmenting some grayscale image or color channel into binary ones and zeros. There are numerous ways to conduct this binarization, as exhaustively summarized by (Sezgin & Sankur, 2004). To limit the scope of this thesis, only a few common approaches will be considered:

In (Efford, 2000), two basic thresholding techniques are given. Both methods, stated in Equation 2.5 and 2.6, are evaluated at every pixel (x,y) according to some predefined thresholds values T, T_1 and T_2. For both techniques, the arrangement of zeros and ones can be reversed for the opposite segmentation result (Efford, 2000).

$$g(x,y) = \begin{cases} 0, & f(x,y) < T \\ 1, & f(x,y) \geq T \end{cases} \tag{2.5}$$

$$g(x,y) = \begin{cases} 0, & f(x,y) < T_1 \\ 1, & T_1 \leq f(x,y) \leq T_2 \\ 0, & f(x,y) \geq T_2 \end{cases} \tag{2.6}$$

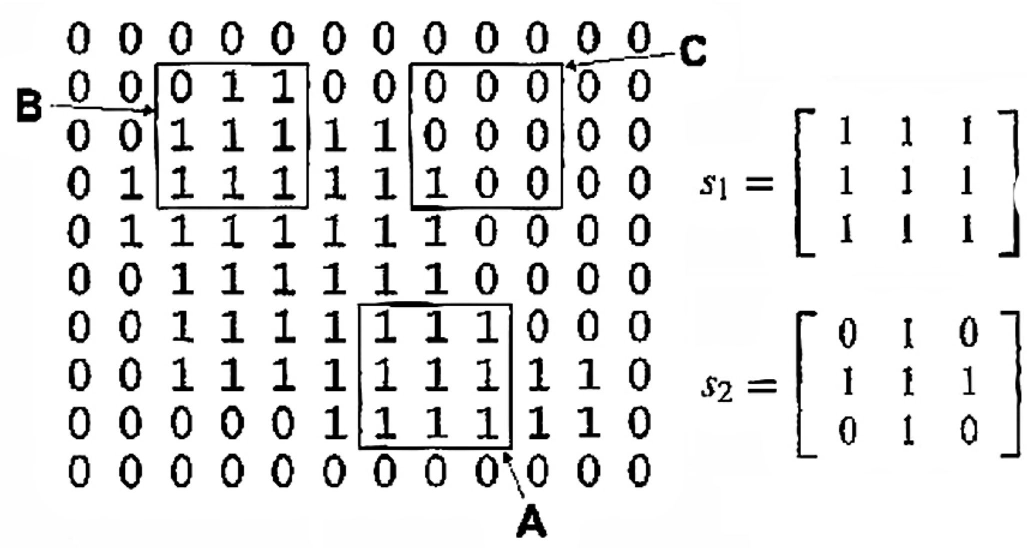
In a constantly changing environment, using static threshold values - as in Equation 2.5 and 2.6 - could potentially compromise the robustness of the target system. On the other hand, these methods are both *global* - they evaluate the whole image according to the same set of rules. This is a necessary feature in for the application of this thesis, as holes and other homogeneous areas potentially could be erroneously segmented by *local* techniques - techniques where each point is evaluated with respect to its neighborhood.

Ideally, the static threshold values would self-adjust with respect to some image indicator. In Sletta's master thesis (Sletta, 2013), a locally adaptive thresholding technique, as well as the optimal, automatic Otsu's Method (Otsu, 1979) were attempted used for net segmentation, but without promising results.

2.8 Morphological Operations

The word morphology can be described as the study of "a particular form, shape or structure" (OED, 2014). In image processing this implies an operation which iteratively evaluates each pixel in an image by overlaying a small template known as a *structuring element* (Efford, 2000). The particular form of this structuring element determines the operation's effect on the target image, and uniquely identifies the morphological operations available. The operations named *dilation*, *erosion*, *opening* and *closing* are commonly used in order to process binary segmentation results, and has been used in combination with background segmentation on several occasions (Cristani et al., 2010)(Cucchiara & Piccardi, 1999)(Cucchiara et al., 2000)(Elgammal et al., 2000). Due to their frequent use and relevance for this thesis, these operations will briefly be described in this section; for detailed information on the topic, the reader is referred to the textbook written by Nick Efford - on which this text is largely based (Efford, 2000).

The mathematical notation for evaluating an image I with a structuring element s can for the operations in this section be denoted as in Table 2.1.



```
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0 0 0 0 0
0 0 1 1 1 1 1 0 0 0 0 0
0 1 1 1 1 1 1 1 0 0 0 0
0 1 1 1 1 1 1 1 0 0 0 0
0 0 1 1 1 1 1 1 0 0 0 0
0 0 1 1 1 1 1 1 1 0 0 0
0 0 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0
```

$$s_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$s_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Figure 2.9: The structuring element $s1$ fit A and hit B and C, while $s2$ fits A and B, with no response to C (Efford, 2000)

Operation	**Notation**
Erosion	$I \ominus s$
Dilation	$I \oplus s$
Opening	$I \circ s = (I \ominus s) \oplus s$
Closing	$I \bullet s = (I \oplus s) \ominus s$

Table 2.1: Mathematical notation for morphological operations

To understand the difference between erosion and dilation, the two terms *hit* and *fit* should be explained: A structuring element, s, is said to *hit* a center pixel (x,y) if at least one binary 1 in s coincide one binary 1 in the neighborhood of (x,y); if all binary ones in s coincide with all binary ones in the pixel's neighborhood, the structuring element is said to *fit* the pixel (x,y). This concept is illustrated in Figure 2.9.

2.8.1 Erosion and Dilation

Erosion is typically used to remove small, unwanted features from some binary image - such as segmentation noise - or to slim the border of some connected set of pixels. A center pixel (x,y) is eroded according to Equation 2.7, as given in (Efford, 2000):

$$g(x,y) = \begin{cases} 1, & \text{if } s \text{ fits } I \text{ at (x,y)} \\ 0, & \text{otherwise} \end{cases} \tag{2.7}$$

Dilation has opposite effect of erosion: Dilation adds extra features, and is commonly used to fill small holes in regions, connect areas that lie close but are separated, or grow the boundaries of regions (Efford, 2000). In background segmentation, dilation is often used in order merge noisy and patchy results into connected objects. The dilation operation is performed for some center pixel (x,y) in the following manner:

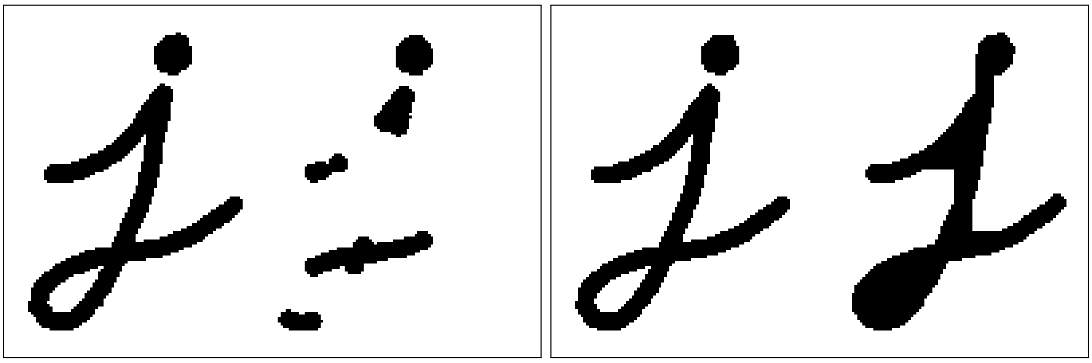

(a) Opening (b) Closing

Figure 2.10: Common morphological operations (Bradski, 2000)

$$g(x,y) = \begin{cases} 1, & \text{if } s \text{ hits } I \text{ at (x,y)} \\ 0, & \text{otherwise} \end{cases} \tag{2.8}$$

On a side note, the *flooding* technique described by (Jakobsen, 2011), can to a large extent be compared to a morphological dilation with a circular structuring element of a defined size - or radius.

2.8.2 Opening and Closing

The name of the opening operation comes from its ability to cut open thin bridges connecting multiple regions. It is a compound operation, consisting of firstly an erosion - in which bridges and small features are removed, but where areas are also shrunk - and then a dilation to bring the areas back to their original size (Efford, 2000):

$$I \circ s = (I \ominus s) \oplus s \tag{2.9}$$

The closing will, as the name implies, close - or fill - areas contained inside some region. The operation consists of firstly dilating the binary image to fill holes, but also grow the outer boundaries of all regions in the image, and then erode the image in order to reduce the scaled regions back to their original size (Efford, 2000):

$$I \bullet s = (I \oplus s) \ominus s \tag{2.10}$$

The opening and closing operations are sometimes preferred over using erosion and dilation directly, since they normally give the desired effect without altering the overall size of all objects in the binary image. An example of the two operators can be viewed in Figure 2.10.

2.9 Evaluation and Research Intention

The main goal of the methodology developed in this thesis is to achieve a robust detection of net damage. It was mentioned in Section 1.3 that this would be approached by firstly separate all foreground elements from some unknown background scene, and then assess the integrity of the net based on the resulting background segmentation image, or binary foreground image.

Several challenges commonly found in the video material analyzed in this thesis were in Section 1.5 introduced. Previously, some of these challenges were solved by (Jakobsen, 2011), (Olsen, 2013) and (Sletta, 2013) using various segmentation techniques based on direct, automatic and optimal thresholding, edge detection, optical flow, hough transforms, and more, but with varying results. For damage detection (Jakobsen, 2011) employed a line search algorithm, while (Sletta, 2013) utilized region growth. Although all of these approaches functioned in multiple scenarios, neither of these were found robust overall; while they mostly managed to detect the net structure itself, complications occurred when, for instance, algae growth or rope structure was encountered.

Karaman found that incorporating multiple segmentation methods, and that employing information for several image channels, typically would improve the robustness of background segmentation systems when used for static background scenes (Karaman et al., 2005). The background segmentation process developed in this thesis will build upon this discovery by combining information from edge detection, motion estimation and temporal background segmentation methods, all of which will be evaluated for multiple image channels, to best ensure a robust segmentation towards both known and unknown challenges. The fundamental assumption of the developed system will be that the weaknesses of each segmentation module will be complemented by the strengths of others.

By considering general properties of temporal background segmentation, this class of background segmentation seems like a generally well suited choice for detecting foreground elements of various appearances, like, for instance, algae growth, which was found problematic by (Jakobsen, 2011), (Olsen, 2013) and (Sletta, 2013). The temporal background segmentation methods that will be studied in this thesis, are those based on a frame buffer structure and a single statistical operator that were suggested by (Lo & Velastin, 2001), (Cucchiara et al., 2000) and (Cucchiara, Grana, Piccardi, & Prati, 2003). It was found in Section 1.5 that the background scenes in the real video material analyzed largely were static and slowly changing, making these per-pixel, monomodal approaches seem feasible. The artificial net setup does not fit this description, however, and would most likely suit a multimodal technique, such as the those found in the GMM family of methods; however, since the focus of this thesis is to develop a system to be used in real ROV operations, this concern is considered low priority, and will therefore be neglected.

The extended S&KB background update approach with motion validated selective update by (Cucchiara et al., 2000), will be extended to also work with edge information. Since no sharp or distinct image points were observed in the background scenes in Section 1.5, including edge information will most likely further improve the selective update, without potentially causing a deadlock scenario.

Methods for edge detection and motion estimation will be utilized both in update scheme of the temporal background segmentation technique, and as standalone modules in the combinatorial segmentation system developed. In (Cucchiara & Piccardi, 1999), the basic motion estimator based on double image differencing purposed by (Kameda & Minoh, 1996), was successfully used for background segmentation. Due to the simplicity of detecting motion with differencing of successive images, a similar approach will be used in this thesis. For edge detection, a novel algorithm will be purposed. This algorithm will utilize the local image sharpness indicator suggested by (Sunkavalli et al., 2012), and the principle behind background subtraction, in order to, literally, subtract all blurred pixels from some analyzed image, which after basic post-processing largely will resemble the result of a traditional edge detector. It should be noted that both the motion estimator and edge detector utilized in this thesis most likely could be exchanged with common, modern equivalents; however, these methods both appear fairly intuitive, and also well suited for a real-time application at some later point.

Both (Jakobsen, 2011) and (Sletta, 2013) found their damage assessment algorithms to function well, but only in certain conditions. By evaluating the binary foreground images resulting form the background segmentation process of this thesis, net damage will be calculated using a series of morphological operations. The aim of this design will be to detect damage without relying entirely on the intactness of the foreground structures in the binary foreground images analyzed, and thereby potentially decrease the rate of false positive detections of damage in conditions where the background segmentation process does not perform reliably.

Statistical filter operators, such as the Gaussian and median filter, will be utilized for various purposes throughout this thesis.

32

Chapter 3

Methods and Materials

In this chapter, the methodology of this thesis will be explained in detail along with the material used. The goal will be to extend the current knowledge of automatic net damage assessment by developing a complete workflow for robust analysis. Several steps of the workflow will incorporate common computer vision techniques and existing research, while specialized solutions will be suggested to handle the particular challenges encountered during underwater net inspection described in Section 1.5.

The methods introduced in this chapter have been developed iteratively through trial and error as different challenges became evident. Therefore, some of the experimental results, as covered in detail in later chapters, will be referenced in this chapter for explanatory purposes.

This chapter is structured in a bottom-up manner, where all methods described eventually lead to the design of the final system introduced in Section 3.11. Other sections of particular interest are Section 3.9 and 3.10, where the combinatorial system design and damage detection method of the final system, respectively, will be explained in detail.

Most approaches investigated and tested in subsequent chapters will utilize some variation of the universal video analysis system illustrated in Figure 3.11, an outline also used in the final system developed. In this outline, the blocks named “Background Segmentation Process” and “Damage Assessment” are of central.

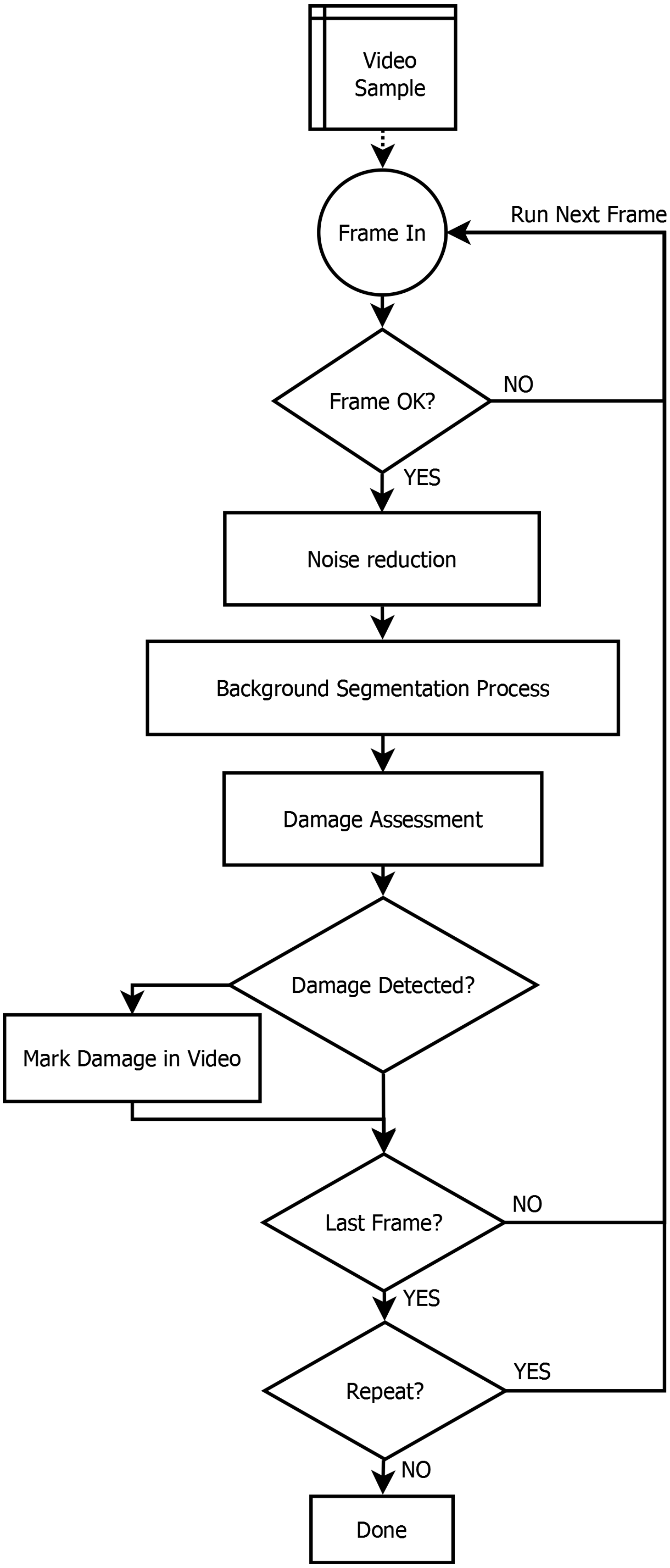


Figure 3.1: Video analysis system outline

3.1 Development Tools

Algorithm development and system tests were conducted in the MATLAB (version: 8.1.0.604, release: 2013a) programming environment on a desktop computer with the following specifications:

Processor	Intel Core i5-2500 3.30GHz CPU
Memory	8 GB
OS	Windows 7 Enterprise SP1, 64 bit
GPU	- not utilized -

Table 3.1: Development Platform Specifications

MATLAB is frequently used for prototyping in the machine vision community, as it allows for quick, orderly code development with well matured support functionality and analysis tools. However, MATLAB is also a scripting language, which makes it unsuitable for most real-time applications and runtime performance testing. The heavily anticipated OpenCV (Bradski, 2000) library in combination with a compiled language such as C++ would have allowed for accurate runtime analysis with a significantly performance boost compared to that of MATLAB. Unfortunately, this combination is not equally suited for rapid prototyping, and would largely have restricted development to pre-implemented library functions. The accessible, low-level algorithmic control available in MATLAB therefore made it the designated tool for this thesis.

Real-time applicability will consistently be deciding factor in the methods utilized in this thesis for two natural reasons. Firstly, the system should preferably be applicable to real-time computation in an untethered vehicle in the future. Secondly, the computer platform specified in Table 3.1 was simply unable to run advanced algorithms in the MATLAB language within reasonable time limits: the least computationally expensive temporal median systems in Section 3.7.1 was computed at an average speed of 4.54 Frames Per Second (FPS) during model tests - 18% of the original frame rate of 25 FPS. Therefore, the development tools themselves naturally limited all tests to computationally inexpensive algorithms.

3.2 Video Test Material

The video material analyzed in this thesis was provided by SINTEF and (Sletta, 2013) and (Olsen, 2013) and consisted of real video footage from an ROV operation near Kattholmen, Norway, and artificially created damage scenario video samples, respectively.

3.2.1 Sample Composition and Highlights

A total of three video samples were created for testing purposes: two of the videos are both based on real ROV inspection video from SINTEF, while the last video combines the multiple test scenarios created by (Sletta, 2013) and (Olsen, 2013). These video clips will be referred to as C1, C2 and C3 for the remainder of this thesis. In

Appendix B.1, an extensive set of reference images highlighting the different situations encountered in each video can be found, while a brief technical summary is given in Table 3.2.

Video	**C1**	**C2**	**C3**
Number of Frames	525	425	582 (930)
Number of Scenes	1	1	11 (16)
Avg. seconds per scene	21	17	2.1 (2.3)

Table 3.2: Background segmentation test video information

Video C1 Highlights

The C1 test video features a single scene from a real ROV operation where the ROV moves slowly in front of the net structure at close range, as illustrated in Figure B.1. The video frames are decently sharp, but offers uneven light conditions and embody most foreground elements encountered during normal operation.

The low level of motion and recurring object position of the foreground elements will challenge the temporal background segmentation methods introduces later in this chapter: unless handled appropriately, the rope structure will make its presence in the estimated background.

This clip does not contain any damage scenarios, and the goal will therefore be to not generate false positives while avoiding excessive oversegmentation.

Video C2 Highlights

The C2 test video features another single scene from a real ROV operation. Contrary to C1, the scene in C2 is captured at a medium range while traversing the net at moderate speeds. Among the clips analyzed in this thesis, C2 best resembles the scenario of a proper net inspection controlled manually by an ROV operator.

Particular challenges found in C2 are: a limited camera resolution due to the medium net range, and motion blur induced by the moving camera platform. All challenges mentioned in Section 1.5 for the real video material applies to this clip as well, most of which are visible in Figure B.2.

Just like C1, this clip does not contain any damage scenarios.

Video C3 Highlights

Clip C3 is radically different from C1 and C2 in several aspects: the video material was captured close to the water surface with a stationary camera using an artificial, motorized net setup. This positioning makes the sun's reflections in the water surface appear as swift lightning changes in the background itself, while also inducing: reflections in the camera optics, bright moving patches in front of the scene, and strong shades in the net structure.

Furthermore, C3 combines multiple separate clips simulating various net tear types. This collection contains both scenes in motion and stationary scenes; scenes with regional focus blur and motion blur; and some regions with net damage and others without. Moreover, some scenes include large semi-stationary objects resembling growth, as well as a wooden frame structure.

A total of 11 scenes with similar light characteristics are incorporated into the error corrected C3, giving 19.4 seconds of video. The resulting quick scene transitions, in addition to the stationary growth structures and rapidly changing light conditions, are the main challenges in C3.

Compared to C1 and C2, this clip truly trials the temporal background segmentation methods introduced later. It also illustrates the weaknesses of several system components and motivates the introduction of multiple additional features. The overall goal of C3 will be to best overcome the presented challenges while accurately detecting the different damage types incorporated into each scene.

In the Figure B.3, prominent frames from C3 are displayed.

3.2.2 Video Conversion and Error Correction

Initially, a format conversion was made on the provided video material, as summarized by Table 3.3, due to incompatibility of the original video formats in MATLAB. While the conversion also unified the crop format, the frame rate stayed the same.

	SINTEF F&A	**Olsen & Sletta**	**Converted**
Codec	MPEG-1/2	MPEG-4 AVC	MPEG-4
Crop Format	1440x1080	1280x720	1200x700
Frame rate	25	30	25/30

Table 3.3: Comparison between original and converted video files

It was noted in Section 1.5 that the video material collected by (Olsen, 2013) and (Sletta, 2013) suffered from a skipping effect. This skipping effect - identified as successive frame duplicates during experiments with motion estimates - were found devastating for the optical flow methods described later in this chapter. Since duplicate frames contain zero relative motion, this sporadic discontinuity made the optical flow estimates completely unreliable for further processing.

Using the motion indicator described in Section 3.8.2, all image frames in C3 were filtered according to their total motion content: if the frame contained less than 1% motion, they were skipped entirely without further evaluation. By removing duplicate frames, the number of scenes included in C3 was reduced from 16 to 11, and the total length reduced from 930 to 540 frames - a reduction of 43%. A graphical comparison of C3 before and after error correction can be viewed in Figure B.9 and Figure B.10, respectively. Further usage of the C3 name will refer to the error corrected video C3.

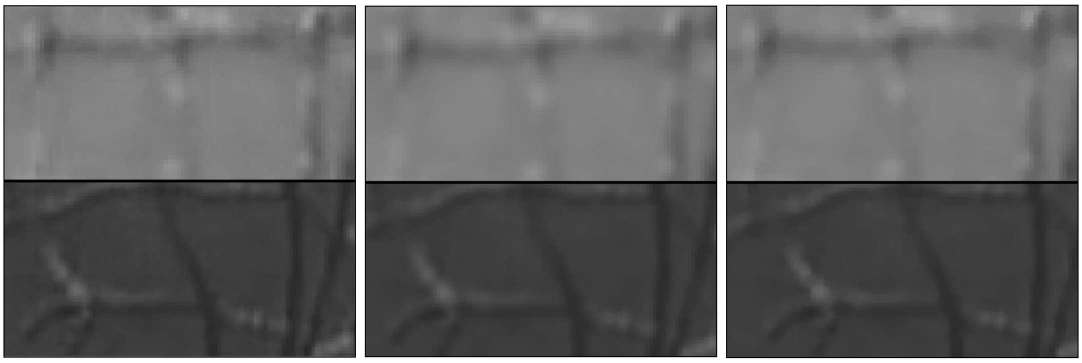
This error management of input frames inhabits the first evaluation block in Figure 3.1.

3.3 Noise Reduction Filtering

In Section 1.5, slight discretization noise was discovered in the analyzed videos streams. This noise can potentially lower the segmentation accuracy of the high resolution video streams by cluttering the background estimate and obscuring image detail. For this reason, a noise reduction step to be executed for all input frames has been incorporated into the main workflow in Figure 3.1.

In (Nixon & Aguado, 2002), a comparison of the Gaussian and median filter was made - as re-illustrated in Figure 2.5. A similar comparison is displayed in Figure 3.2 on images collected from C1 and C3, where both the median and Gaussian filters reduced the discretization noise of the original image. Although barely visible, the Gaussian filter smoothen the background regions slightly better, while the median retains more image detail. The template size T_s and standard deviation σ were picked manually, aiming to best smoothen the background without loosing image detail.

In the workflow of Figure 3.1, the Gaussian filter will be incorporated due to its better ability to smoothen background regions. With the backgrounds being largely homogeneous, this effect is believed to give a better background estimate and subtraction result, overall improving segmentation performance.



(a) Original, blue channel (b) Gaussian, $\sigma = 3$, $T_s = 3$ (c) Median, $T_s = 3$

Figure 3.2: Noise filter comparison of blue color channel on close-up image of the two scenes in Figure 1.4d (lower) and 1.4e (upper)

3.4 Image Subtraction and Thresholding

Image subtraction - or differencing - refers to the action of literally subtracting one image from another, resulting in a new image containing the value differences for each pixels between the two frames. In its simplest form, the resulting image may contain both positive and negative values, as for Equation 3.1, also illustrated with a histogram representation in Figure 3.3. Because of this property, the particular order of subtraction will affect the differencing result, a vital element when combined with some thresholding techniques - which will be discussed next.

$$I_{Diff} = I_A - I_B \tag{3.1}$$

The thresholding method that will be used throughout this thesis is given in Equation 3.2, and resembles the binarization method in Equation 2.6 described by (Efford,

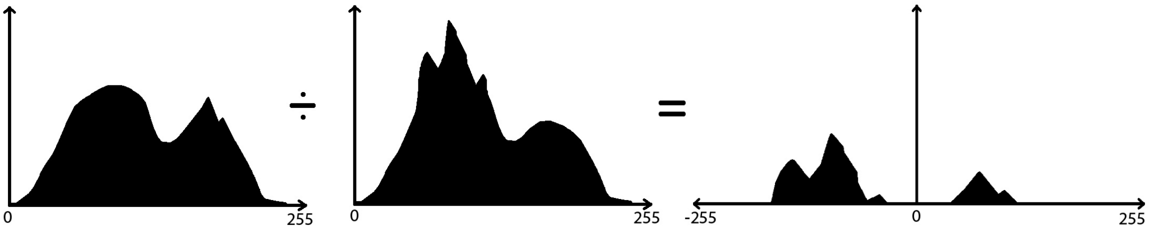


Figure 3.3: Image histograms before and after image subtraction

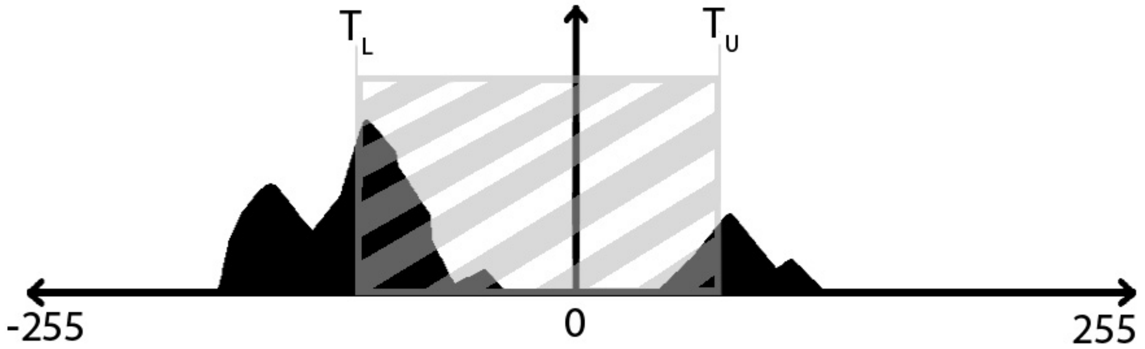


Figure 3.4: Double direct thresholding of difference image

2000) with some slight variations. As seen in Figure 3.4, this binarization method, which will be referred to as Double Direct Thresholding (DDT), bears close similarity to a band-stop filter for image histograms, where the upper and lower threshold values, T_U and T_L respectively, defines the bandwidth.

$$g(x,y) = \begin{cases} 1, & f(x,y) \leq T_L \\ 0, & T_L < f(x,y) < T_U \\ 1, & f(x,y) \geq T_U \end{cases} \tag{3.2}$$

Compared to the simpler thresholding method given in Equation 2.5, DDT allows for additional segmentation control. For instance, when used in combination with temporal background segmentation - such as Figure 2.7 - certain types of image elements might appear mainly negative or positive in the difference image. If DDT is used, one may specifically choose to segment only some image elements, whereas the simpler thresholding technique would evaluate all elements according to equal terms, potentially suppressing image information in the process.

The choice of using a static thresholding technique like DDT does not work according to the plan of developing an automatic, robust inspection system that handles various scenarios without user interference. During early experimentation with thresholding in combination with temporal background segmentation, it was found that neither adaptive thresholding nor Otsu's Method performed as desired. The methods were in particular tested on the positive and negative histograms of the difference images separately - a scenario not previously investigated by (Sletta, 2013). The results were usable at best, but highly unstable - as similarly concluded by (Sletta, 2013). The DDT method with manually chosen thresholds was used for comparison, giving superior segmentation accuracy for all test instances. Although not ideal, DDT will therefore be used in order to limit the scope of this thesis. However, in the interest of finding an adaptive parameter scheme through further studies, the thresholds found will be logged in detail.

For all algorithms incorporating the DDT binarization technique, the aim will be to find *uniform* parameter settings. Uniform in this context, will refer to a single set of static threshold settings, unique for each color channel, that works sufficiently for all video material analyzed - the clips C1, C2 and C3 - when combined in a specific multi-channel combinatorial methodology that will be introduced in Section 3.9. In contrast, *unique* parameter settings will refer to thresholds specifically picked to best function for each individual image channel and each individual video clip, without concerning about robustness of the parameter sets to changing scenes. As will be seen later, a uniform, combinatorial approach will allow some methods to function robustly in multiple scenarios using only a single set of static DDT parameters without significant loss of accuracy.

3.5 Optical Flow Motion Estimation

Based on the double differencing motion estimation workflow developed by (Kameda & Minoh, 1996), as illustrated in Figure 2.8, two procedures have been developed. The first method is a direct interpretation of Kameda's workflow where the binarization has been exchanged by DDT, and the clustering step has been moved: a morphological closing will be applied at a later point in the system. The second method is a simplification of the first, where only a single differencing image is used for motion estimation. Both methods are illustrated in Figure 3.6, where they have been named *single* and *double* differencing accordingly.

The main distinction between the single and double differencing methods, is how they react to objects of different sizes moving at different speeds, or specifically, whether these objects overlap between successive frames or not. These characteristics have been illustrated with an example in Figure 3.5, where a black ball passes through some image region, with a slight overlap between each frame. In this illustration, notice how the double differencing method always will lag one iteration behind the current frame k, which is why the last estimate $k = 4$ for the double differencing procedure in Figure 3.5 is crossed out.

The single differencing method has been developed with the intention of better detecting the complete, continuous structure of small objects in motion. In terms of the example in Figure 3.5, it is visible how the size of the detection always will be larger than the object itself. Furthermore, if the moving object overlaps between frames, the overlapping area will not be detected. If the object is slowly moving, this overlap might be significant, eliminate a major part of the detection; however, when post-processed with a morphological closing operation this eliminated region will most likely be filled, reducing the issue. Since the pixel size of the overlap will decrease for smaller objects, the closing operation is also more probable to fill the detection gap for such objects. Moreover, as seen in Figure 3.5 the single differencing detection area extends beyond the size of the moving object by generating a tail in the location of the previous motion estimate. While this tail potentially can result in excessive or duplicate segmentation for quickly moving objects, it might also cover immediate

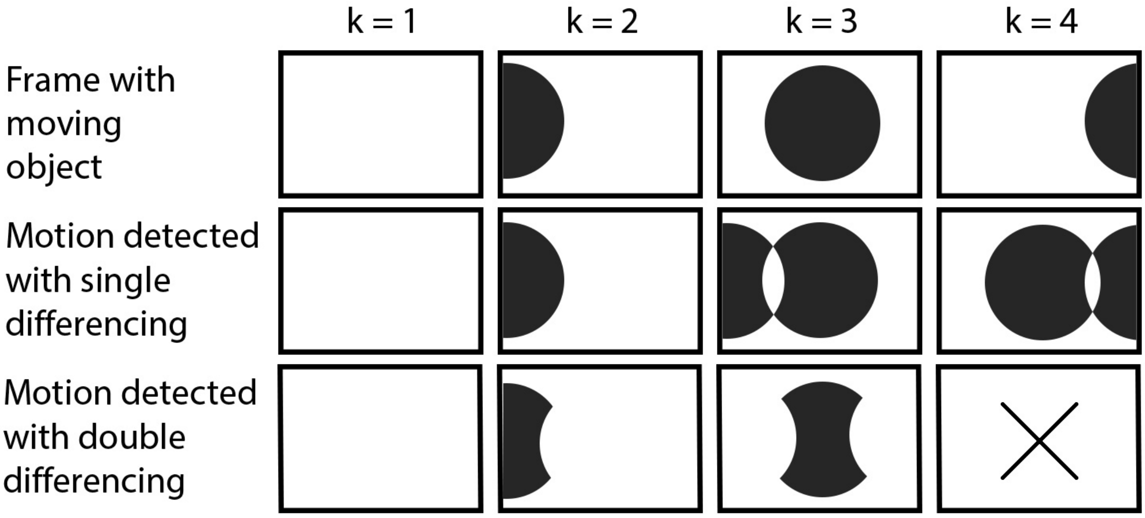

k = 1
k = 2
k = 3
k = 4
Frame with moving object
Motion detected with single differencing
Motion detected with double differencing

Figure 3.5: Principal difference between single and double differencing

neighboring pixels during moderate motion, improving the overall detection.

Where the single differencing method extends the detection area by generating a motion tail, the double differencing does not; while the binary AND operation will eliminate overlapping object structure between frames, it will also remove the tail and possible duplicate segmentation. For thin structures, such as net threads, moving at slow speeds, this overlap removal might compromise the integrity of the continuous net structure, or even remove the structure completely. And since the double differencing eliminates the tail of the single difference images, a morphological closing or dilation operation might not have any remaining pixels to operate on, leaving the structure undetected.

A general trait of both differencing motion estimators, is their ability to detect spontaneous brightness changes, as could occur from solar reflections in the camera optics, or similar, as discussed in Section 1.5. In terms of maintaining a stable background estimate for a temporal background segmentation method incorporating an S&KB selective update scheme, which will be discussed later, detecting such scene changes might be beneficial.

Optical flow estimation by image differencing will be used as a component of the S&KB update scheme discussed later in Section 3.7.1; as a standalone segmentation module tested for both a unique and uniform design schemes, as described in Section 3.9; and further processed to work as an image motion indicator in Section 3.8.2. Neither of these purposes require information about the precise movement of individual pixels or pixel regions, which makes implementing a more advanced and computationally complex optical flow algorithm with tracking abilities largely redundant.

Most important for the S&KB update scheme is a detection of motion that covers the entire structure of the foreground elements, so that pixels estimated to be in motion can be excluded from the background update, regardless of their previous position. The excessive detection of the single differencing technique might be a suitable selection for this task, but the double differencing procedure might also work, depending on the amount of motion present and the size of the foreground objects. Since most scenes analyzed in this thesis contain thin net structures and large foreground objects at close and long range, which change between being semi-stationary and quickly moving, the motion estimate will have to tolerate all cases. While the single differencing procedure most likely will detect the foreground elements in all cases, albeit with the possibility of excessive or duplicate segmentation, the double procedure might not be able to detect slowly moving net structure.

The image motion indicator will measure the amount of relative motion. So as long as the amount of pixels in motion scale with the amount of movement present in the image, any optical flow estimate will do.

Originally, the single and double differencing methods were both investigated through the work of this thesis; however, at a late point in the work process, a subtle error was discovered in the implementation of the double differencing procedure, which made the initial arguments of not utilizing the double differencing estimator based on the experiments conducted invalid. Due to this unfortunate incident, the double differencing procedure will not be documented in this thesis, while the single differencing estimator, originally found to be better, will be utilized. After reworking the double differencing implementation, its performance improved drastically, making it a real contender of the single procedure. For further work, it might therefore be worth investigating.

The image subtraction step in Figure 3.6 for the single differencing procedure, will be conducted in a descending order, that is $I_k - I_{k-1}$.

The single differencing optical flow estimation technique will be tested firstly with unique DDT parameter selections for each scene and image channel, results of which will be used to find uniform parameters intended to function for all scenes in a multi-channel combinatorial approach. Exact implementation details for the unique and uniform approaches will be introduced in Section 3.9.

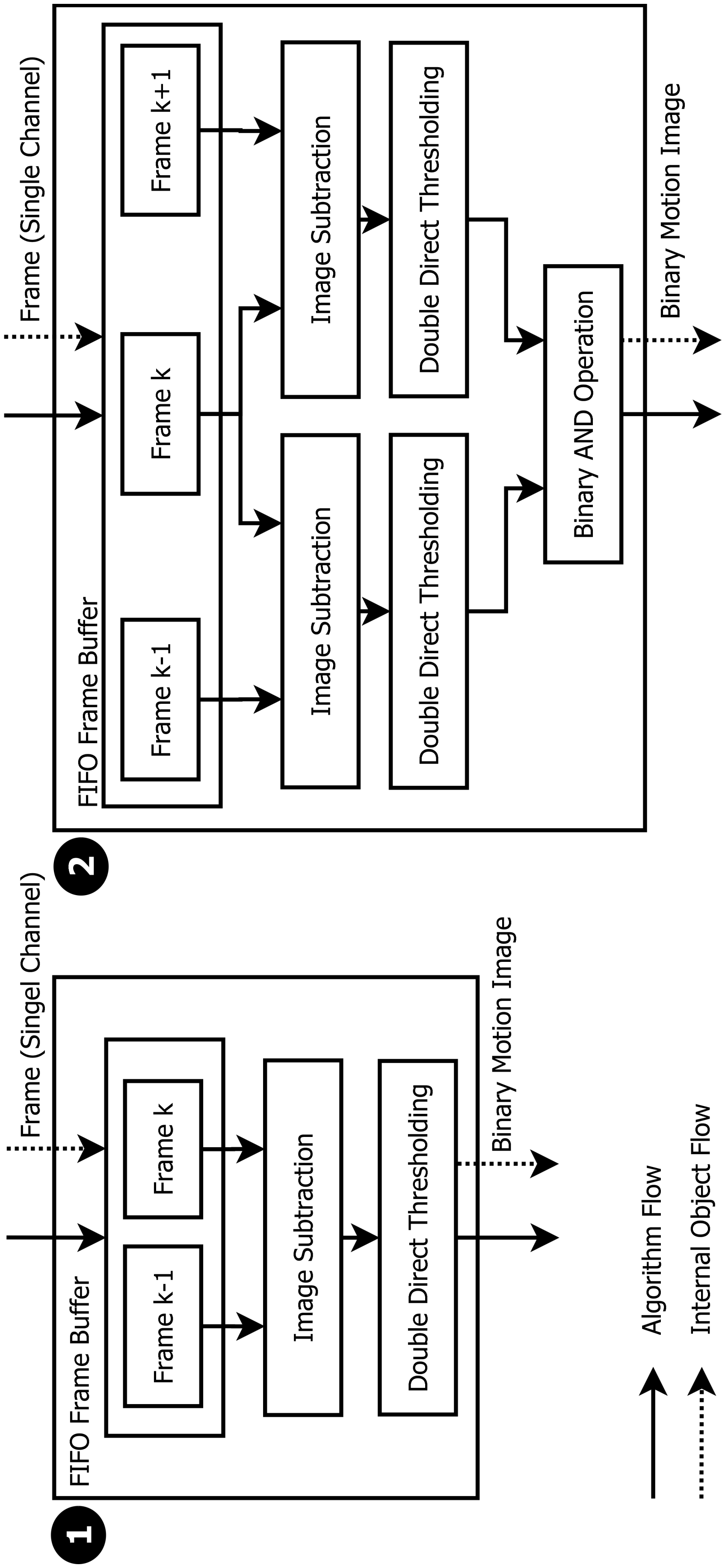


Figure 3.6: Optical flow motion estimation through: **1.** Single Differencing, **2.** Double Differencing

3.6 Local Sharpness Point Detector - A Novel Edge Detector

A novel low-level feature detector, inspired by the local image sharpness measure presented by (Sunkavalli et al., 2012), will here be proposed. The complete algorithm - namely the Local Sharpness Point Detector (LSPD) - is displayed in Figure 3.8, from where the two first blocks can be formulated mathematically as in Equation 3.3. The methodology of this procedure, which now will be explained, is illustrated in Figure 3.7.

$$S_k = I_k - G_\sigma \otimes I_k \tag{3.3}$$

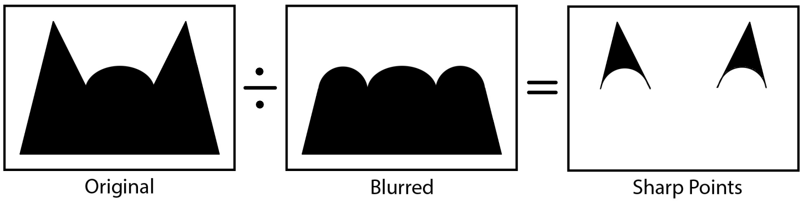


Figure 3.7: Local sharpness point detector methodology

By subtracting an image I_k by its blurred equivalent $I_k \otimes G_\sigma$, the result will only contain pixels that are not equally blurred in the original image I_k, as seen in Figure 3.7. Image points of high contrast, such as edges and image noise, will therefore get a non-zero sharpness value S_k - relative to their contrast level - suitable for binarization through thresholding. Contrary to image edges, noise pixels will have a widely scattered distribution across the binary image; by applying an appropriately sized median filter this noise can therefore largely be removed. The final binary image will contain points of distinctively high relative, local contrast, such as visually ”sharp” image points - or edges.

Theoretically, the LSPD has good localization and single response properties compared to many conventional edge detector algorithms: points detected by the LSPD does not deviate from their original edge position, and single edge points will only give a single binary response. The noise suppression and detection rate, on the other hand, depends entirely on the parameter settings chosen. Furthermore, since the local, relative contrast of image pixels is utilized in the LSPD, it will most likely be insensitive to lightning changes.

Naturally, the LSPD is limited to detecting sharp image points, and will therefore not detect smooth image objects very well. For this reason, its aim will be to isolate the net structure from the background. Soft objects, such as clusters of algae growth, the background, and regions out of focus will most likely yield a low response.

The LSPD will be utilized for edge detection in this thesis, with the goal of improving the selective filtering ability of the S&KB update scheme described in Section 3.7.1 by complementing the double differencing motion estimate for low motion situations. Moreover, the LSPD will be used as a standalone segmentation module for detecting continuous net structures. Just like the single differencing optical flow algorithm, the LSPD algorithm will be tested with both a unique and uniform combinatorial component designs, which will introduced in Section 3.9.

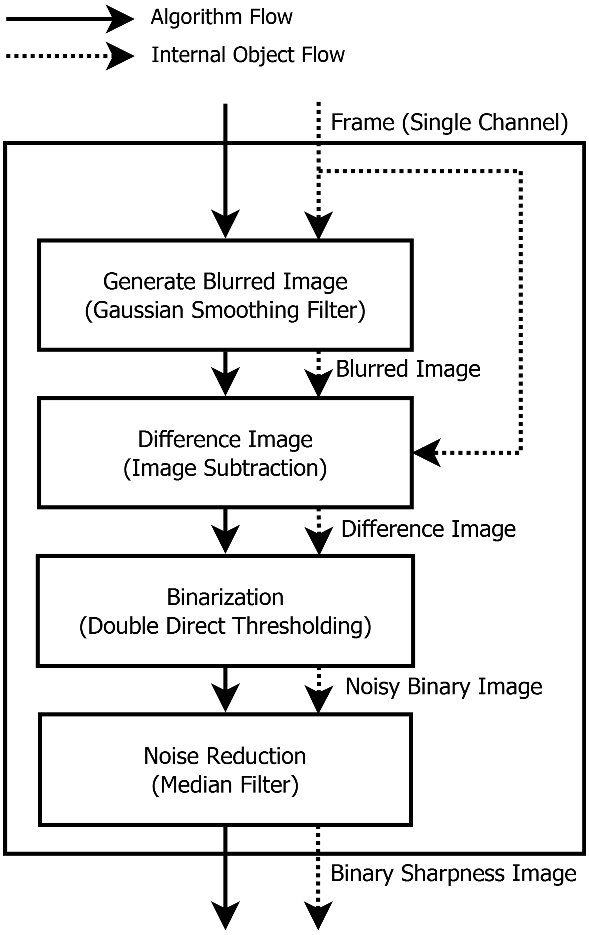


Figure 3.8: Local sharpness point detector workflow

3.7 Temporal Background Segmentation

It was concluded in Section 1.5 that there was little relative motion between the camera and the background, and that the background scene itself was largely homogeneous and slowly changing in the realistic video material analyzed. Based on these evaluations, a temporal background segmentation approach for isolating foreground objects seems feasible.

A major benefit of temporal background segmentation methods is their ability segment any object that somehow distinguishes itself from the background if an accurate background estimate can be calculated and updated. If the estimate is erroneous or outdated, however, temporal background segmentation methods may potentially classify all image pixels as foreground, or oversegment large image regions. When utilized, this class of segmentation methods might therefore require careful maintenance and consideration.

In Figure 3.9, a simplified subsystem module for the temporal background segmentation methods investigated in this thesis is illustrated. If the outline in Figure 3.1 were to use a single image channel temporal background segmentation method, this module would replace the “Background Segmentation Process” - component of this workflow.

In Figure 3.9, every m frame will be trajected to the background model for evaluation, where $m = \triangle t \cdot T_f$ is a positive integer value that will be used as a measure of the sampling interval relative to the frame rate of each video. In a practical implementation, m would represent the frequency of which a sample is selected for analysis from a continuous video stream; for instance, if $m = 10$, then every 10th frame would be analyzed, independent of the frame rate of the analyzed video material. In order to avoid confusion, the relative sampling interval, m, will further be referred to as the *sampling frequency.*

Furthermore, the background subtraction operation in Figure 3.9 will be evaluated in the following manner:

$$I_{Diff,k} = I_k - B_k \tag{3.4}$$

where B_k is the background estimate, I_k the input frame, and $I_{Diff,k}$ the resulting difference image at index k. All temporal segmentation methods investigated will use the DDT binarization technique with manually selected thresholds, in order to fully control the segmentation results.

The methods purposed will be evaluated according to how well the following criteria are met:

1. Fast model initialization and learning rate of an obstructed, unknown, changing scene
2. Exclusion of all foreground elements during model update
3. Robust segmentation of all objects not belonging to the background scene
4. Real time processing capability

Table 3.4: Evaluation criteria of the temporal background segmentation methods

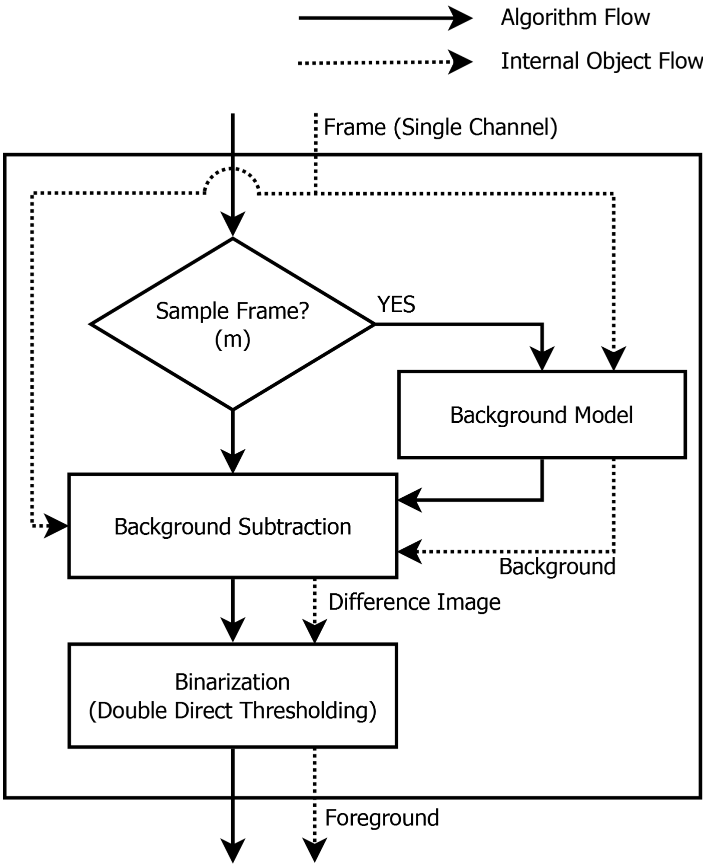


Figure 3.9: Temporal background segmentation general workflow

Since the video clips C1 and C2 contain realistic video capture, fulfilling these criteria for C1 and C2 will be prioritized. The rapidly changing artificial nature of C3 largely opposes the assumption made about stability and slowness of the background scene, and the criteria in Table 3.4 might therefore not suit the background models investigated, which will be taken into account during performance evaluation.

3.7.1 Background Models

A total of three temporal background estimation models will be evaluated in this thesis, namely:

1. Temporal Median Operator w/ Blind Update (TMBU)
2. Temporal Median Operator w/ S&KB Selective Update (TMSU)
3. Temporal Median Operator w/ Combinatorial Selective Update (TMCSU)

The TMBU and TMSU models largely resemble the background segmentation methods purposed by (Lo & Velastin, 2001) and (Cucchiara et al., 2000), respectively, applied to the current application, while the TMCSU system has been specifically designed for the purpose of ROV net inspection, and combines the methodologies and results from multiple sources and experiments.

Neither the TMBU nor the TMSU models will be implemented in the final system, but have been included for one particularly important purpose: the experimental results from these models will trial several techniques later to be used in the TMCSU model and final system. By conducting these experiments in an isolated environment with as few parameter choices as possible, the effect of each technique and parameter can be securely determined. In a complex, combined system, this would not always be possible. As such, experiment and component test results will be administered in the following manner:

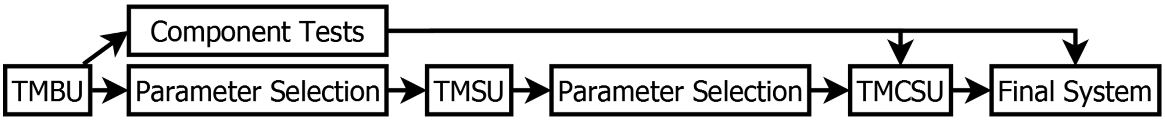


Figure 3.10: Background model development flow

The statistical median operator was chosen for all three background models, as it was concluded less computationally expensive and more accurate than the average and mode operators for real-time evaluation by (Cucchiara et al., 2000). These considerations seem to suit the evaluation criteria in Table 3.4 well.

A schematic overview of the TMBU and TMSU models with their shared parent module can be seen in Figure 3.11. Similarly, the TMCSU model and its parent module is visible in Figure 3.12.

Before each model is discussed, the terms *learning rate* and *initialization period*, and the operation of the Expanding FIFO Buffer utilized in all models, will be defined.

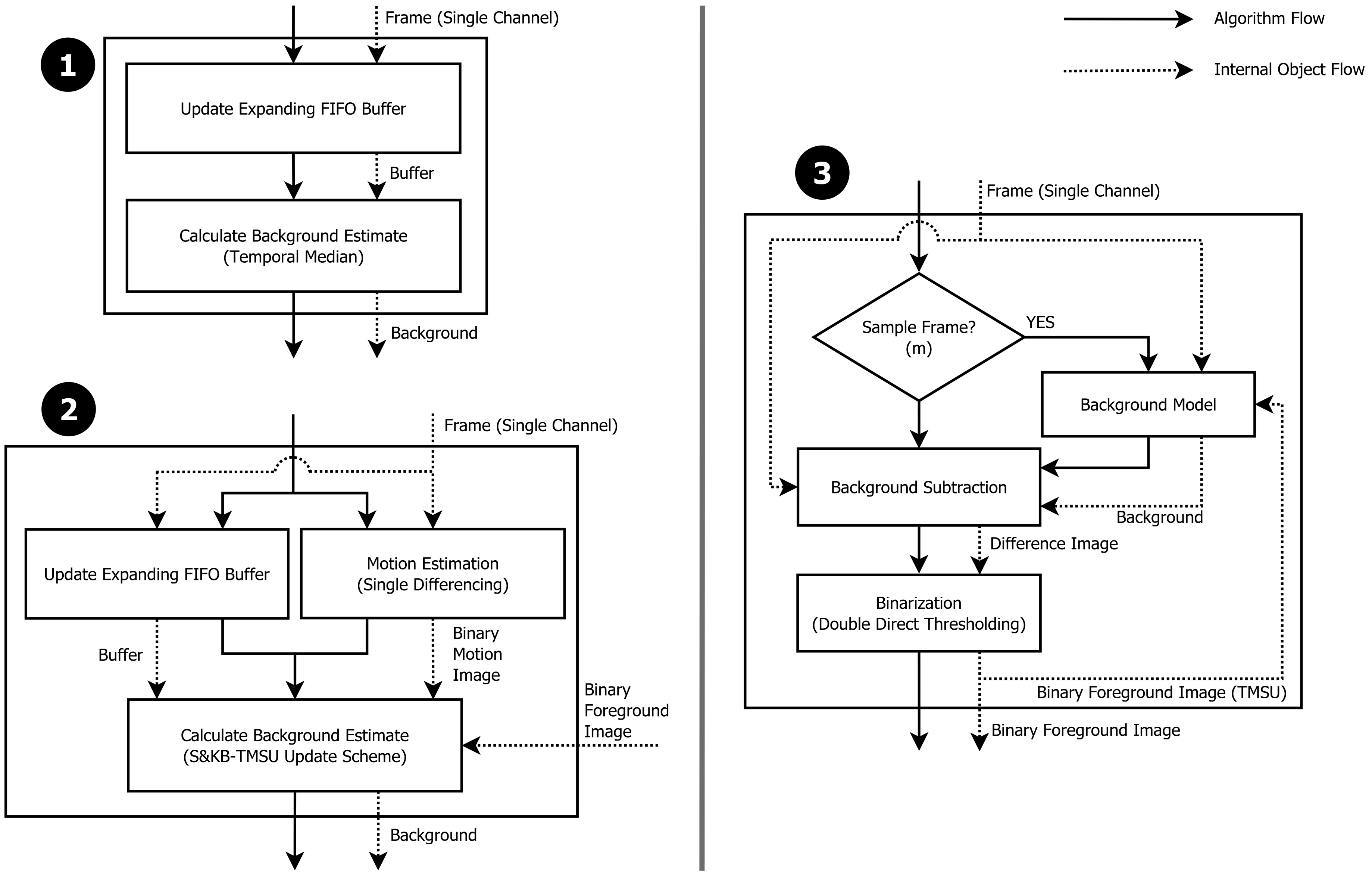


Figure 3.11: Background Update Models: **1.** TMBU, **2.** TMSU, **3.** Parent module

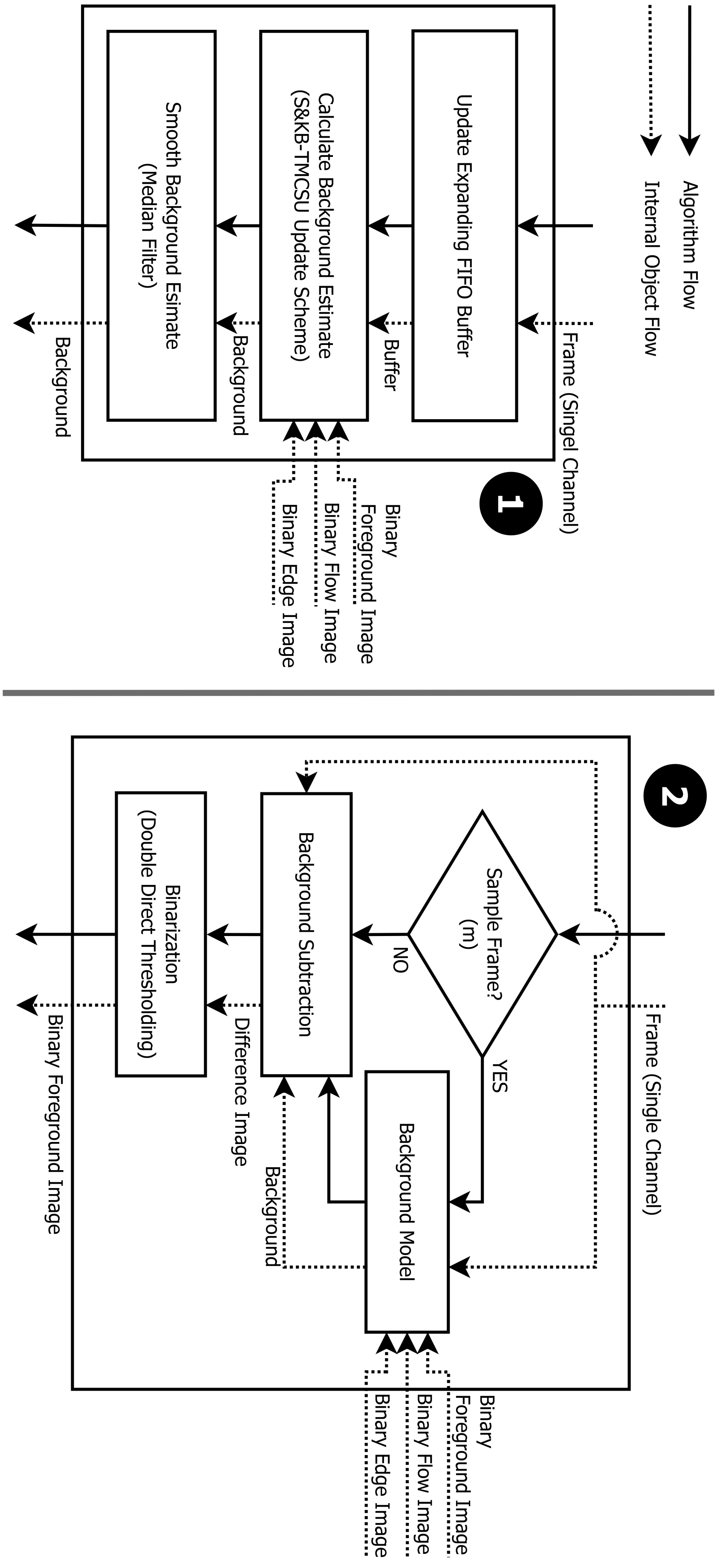


Figure 3.12: Background Update Models: **1.** TMCSU, **2.** Parent module

Learning Rate and Buffer Initialization

A background model's *learning rate* (LR) is here defined as its ability to adapt to permanent changes in the scene, such as varying light settings, image gradients or moving objects that become stationary. A model will be assumed fully adapted when the rate of segmentation stabilizes to normal levels after some change to the background scene.

The *initialization period* (IP) will represent a model's ability to fully define a background estimate after a system reset or startup. Due to the selective update scheme utilized in the TMSU and TMSCU models, not all pixels in the background estimate will immediately be assigned a value, appearing black (of zero value) in the calculated estimate. For all background models, undefined pixels would be a concern if the buffer structure were to be initialized empty and of fixed size; the Expanding FIFO Buffer structure will provide a gentle buffer initialization, avoiding this issue.

The Expanding FIFO Buffer structure is very simple manner of providing a smooth buffer initialization during startup and system reset. At initialization, its size will be zero, but as frames are fed into the background model, the buffer will grow to a predefined maximum size. When the maximum size is reached, the buffer will organize frames in a FIFO manner - just like a fixed size buffer. The benefit of this expanding buffer initialization, is that the background model will provide a functional estimate from the very first frame, whereas a fixed buffer would require at least half of its capacity to be filled before the median operator would calculate a non-zero background estimate.

Both the learning rate and initialization period will be measured in the number of frames required for scene adaption or background initialization, respectively. In some of the experiments conducted later, however, these will both described with the graphs from the frame indicators in Section 3.8.

Temporal Median with Blind Update

The TMBU background model visualized in Figure 3.11, is an interpretation of the temporal median background subtraction scheme suggested by (Lo & Velastin, 2001), as presented in Figure 2.7, with some additional features; the TMBU model incorporates an expanding buffer, and the parent system utilizes a different input filtering and binarization technique. This is the most basic background model investigated in this thesis, which has been included for several experimental purposes:

The TMBU model relies only on two parameters: the sampling interval $\triangle t$ and (maximum) buffer size n. This makes it suited for determining the effect of these two parameters, both of which will be deciding factors with the TMSU and TMCSU models later, in terms of stability, smoothness and computational complexity of the background estimate.

If the background scene is assumed slowly changing and homogeneous, a TMBU model with an extremely large buffer would most likely give a very good indication of what the ideal background estimate would look like. This estimate would then be well suited as a source of comparison for the TMSU and TMCSU background models. The final system will, however, not embed such a model, as an extremely large buffer would most likely implicate extreme memory requirements, as pointed out by (Piccardi, 2004), and also a non-existent learning rate.

The background estimates found from the TMBU model experimentation will be used in order to determine the isolated effect of several individual system components later to be embedded into the final system and TMSU and TMCSU models. Specifically, an extensive analysis of threshold values for the DDT technique will be conducted, which then will allow for morphological operations to be analyzed. Furthermore, the effect of background smoothing before binarization will be investigated, and the possible benefit of combining image channels studied. In general, since the TMSU and TMCSU background models both are derived from the TMBU model, experiments with this model will allow for an in-depth analysis of the most fundamental properties of the TMSU and TMCSU models as well.

Since the TMBU model utilizes a blind update scheme, both the initialization period and learning rate of this method will be predetermined by the buffer size, n, and sampling frequency, $m = \triangle t \cdot T_f$, that is:

$$IP_{tmbu} = n \cdot m \tag{3.5}$$

Temporal Median with Selective Update

As an iteration of the TMBU background model, the TMSU model implements a selective update scheme aiming to protect against erroneously included foreground objects in the background estimate by evaluating optical flow. This improvement is designed to achieve the background estimation accuracy of a TMBU model with a very large buffer structure while using a buffer of limited size to offer a real-time compatible computational complexity, a short initialization period and a high learning rate.

The update scheme of the TMSU model - the S&KB-TMSU update scheme - utilizes a modified version the S&KB method proposed by (Cucchiara et al., 2000). For optical flow estimation, the single differencing technique is used, as seen in Figure 3.11. The simplicity of the single differencing technique, and its potential to provide a continuous binary motion image, makes it seem like a suitable choice.

The S&KB-TMSU background update is stated in Equation 3.6:

$$B_k = \begin{cases} B_{k-m}, & \text{if } I_k \in \{F_k^p = 1\} \\ Median(I_k, I_{k-m}, \, ... \, , \, I_{k-(n-1)m} \, , \, w_b B_{k-m} \,), & \text{otherwise} \end{cases} \tag{3.6}$$

where F_k^p is the binary *update blocking filter* evaluated for every pixel $p = (x, y)$; m is the sampling frequency; w_b is the number of times the previous background estimate B_{k-m} is appended to the buffer during evaluation, referred to as the *background preservation rate* from here; n is the expanding buffer maximum size; and I_k, B_k and k are the current input frame, background estimate and frame index, respectively.

The update blocking filter, F_k, is an abstraction of the binary and thresholding operations embedded into the original S&KB statement in Equation 2.2 by (Cucchiara et al., 2000), which here will be managed as a separate system component. It functions as a frame overlay, where all pixels in the input frame covered by a binary one in the blocking filter will be ignored during background update. In Figure 3.13, the blocking filter for the TMSU model update is presented. Apart from the morphological operations, this is a direct interpretation of the original S&KB statement visualized in a flow diagram. The morphological closing operations have been included in order to achieve compactness of the binary object structures, aiming to improve the robustness of the selective update by assuming that neighboring pixels share the same origin.

Experiments conducted with the TMSU background model will focus on documenting the effects of the selective update by varying the background preservation rate, w_b, in combination with the buffer size and sampling interval, in order to find a working setting for the TMCSU background model. Foreground binarization and single differencing threshold, as well as morphological closing settings, will be assumed found by prior to these experiments. Ideally, the TMSU model will be able to estimate the background scene accurately without including foreground structures into the background estimate.

Temporal Median with Combinatorial Selective Update

The TMCSU background model offer a set of improvements over the TMSU model formulation. The key points of difference can be summarized as follows:

1. Edge information embedded into the update blocking filter
2. Smoothing of background estimate
3. Multiple image channel support

The TMCSU model's background update scheme is identical to that of the TMSU model, but with information from edge detection incorporated into the calculation of the update blocking filter. This scheme, namely the S&KB-TMCSU update scheme, is formulated mathematically by Equation 3.6, where the blocking filter, F_k, is calculated according to Figure 3.13.

Edge information has been introduced into the blocking filter in order to provide additional robustness for situations where the motion estimate might prove inadequate in isolating foreground elements. For the selective update to work reliably with optical flow estimation alone, as in the S&KB-TMSU scheme, the foreground must be in constant motion relative to the camera, or otherwise erroneously influence the background estimate. It was discovered in Section 1.5 that foreground indeed usually is in relative motion to the camera, but not always. To ensure robustness of the background estimate when there is no or little motion information available, one may utilize the observation that the background scene in fact contains no distinct objects, but is rather largely homogeneous and smooth. An edge detector algorithm will not respond to a smooth and homogeneous background, and therefore not erroneously exclude background areas from the update. Most foreground objects, however, are highly detailed and of distinctive appearance, and will therefore respond well to an edge detector algorithm. As a result, by combining the foreground segmentation estimates from optical flow and edge detection, one may improve the overall stability of the selective update scheme without compromising the learning rate of true background pixels, avoiding the deadlock scenario discussed in Section 2.3.2. Furthermore, an edge detector and optical flow estimator naturally complement each others weaknesses: whereas the an edge detector might struggle when exposed to motion due to blur, an optical flow estimator will thrive with increased relative motion; and reversely, an optical flow estimator will respond poorly with no motion present, while the resulting increased image quality will boost the response of the edge detector.

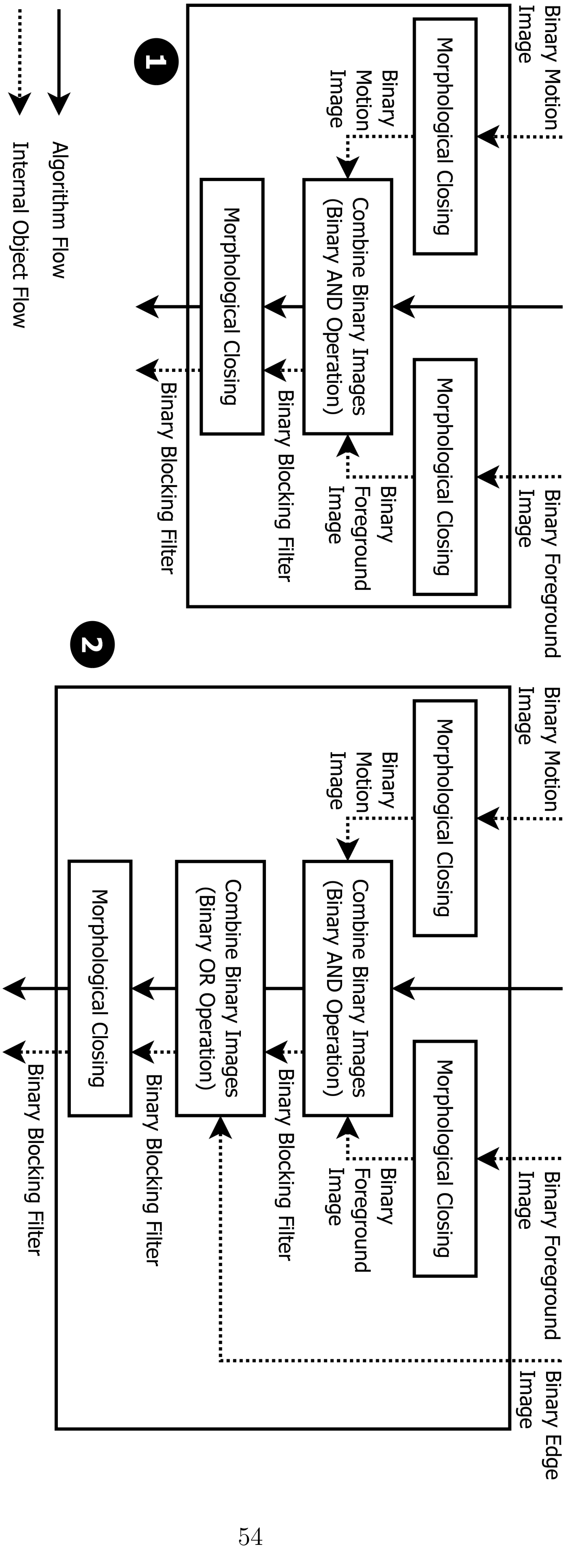


Figure 3.13: Update blocking filter calculation for: **1.** S&KB-TMSU , **2.** S&KB-TMCSU

The edge detector also has its limitations, and the methodology described above will therefore not ensure robustness of the background estimate in all situations. Image elements that belong to the foreground but are of a smooth character, such as algae growth or image regions severely out of focus, will most likely be classified as background both by the edge detector and optical flow estimate when there is little relative motion present. This is a challenging scenario, as there are few characteristics that can be used to uniquely identify such smooth foreground objects from background pixels, without, for instance, analyzing texture patterns or relative composition of brightness or color. An approach that could avoid the issue altogether, would be to lock the background update completely if the ROV was found to be stationary or no relative motion was detected. Such *motion locking* would require some reliable, scene-independent estimate of relative motion, either by analyzing optical flow, or through integration with the ROV's internal sensory equipment. Motion locking was, in fact, incorporated into an early prototype of the S&KB-TMCSU update scheme, where the motion indicator in Section 3.8.2 was utilized as a decision variable, but with mixed results: the approach handled sporadic stops of the ROV well, such as when it was changing direction of motion, but also severely crippled the learning rate and initialization period of the background model in certain cases. Moreover, locking the background update in low motion scenarios largely counteracts the strengths of the edge detector for images with low motion blur. Ultimately, motion locking was not implemented into the S&KB-TMCSU update scheme, as the initial challenge was found less degenerative when a uniform multi-channel system design was implemented, as will be introduced in Section 3.9.

An inevitable side effect of reducing the buffer size of a background model that incorporates a median operator, as will be documented in Section 4.5, is the increased roughness of the background estimate. This roughness, which is of distinctive but distributed nature, can possibly result in the erroneous segmentation of background pixels, or segmentation inversion, potentially reducing the quality of the temporal segmentation result. However, if the true background scene indeed is smooth, then smoothening the background estimate will most likely make it more accurate, reducing this issue. For this reason, the TMCSU model in Figure 3.12 employs a median smoothing filter. Although the Gaussian filter was appointed as the optimal smoothing filter in (Nixon & Aguado, 2002), the similarity to salt and pepper noise of the roughness of the background estimate makes the median filter appear as the better choice. Furthermore, since the S&KB-TMCSU update scheme not necessarily will initialize all pixels in the background estimate simultaneously due to the selective update methodology, the median operator might also reduce the degraded segmentation accuracy during buffer initialization occurring from undefined (black) pixel values.

The TMCSU model has been designed to receive binary foreground, optical flow and edge information from an external source, as seen in Figure 3.12. This feature allows it to utilize information combined from multiple image channels, and standalone segmentation modules. In Section 3.9, two such combinatorial system designs will be presented, namely the unique and uniform combinatorial system designs. For external edge detection and optical flow estimation, the LSPD and single differencing methods will be used, respectively. As will be seen later, these techniques complement each other and the temporal background segmentation well in that they segment foreground objects of different types, overall improving the accuracy of the update blocking filter.

They are also both computationally fast.

3.8 Frame Quality Indicators

In order to better analyze and evaluate the component and system tests conducted the following chapters, quality indicators for determining sharpness and motion content of the current frame, initialization amount of the background estimate, and segmentation amount of the foreground, has been developed. All of the indicators are designed to provide a percentile measure, while also managing the previous Q estimates in a running mean calculation as a historical reference. The running mean and quality estimate was calculated according to Equation 3.7 and 3.8, equally for all indicators.

$$Mean_k = \begin{cases} \frac{Mean_{k-1}(k-1)+Estimate_k}{k} & , \quad \text{if } k < Q \\ \frac{Mean_{k-1}(Q-1)+Estimate_k}{Q} & , \quad \text{if } k \geq Q \end{cases} \tag{3.7}$$

$$Estimate_k = \frac{100}{width \cdot height} \cdot \sum_{\forall(x,y)} U(\cdot) \quad [\%] \tag{3.8}$$

where the $U(\cdot)$ is a function returning some binary image.

3.8.1 Sharpness Indicator

Measuring the amount of sharp points in the tested videos is of interest, as it can be used as a general indication of image quality, while also functioning as a reference for the amount of foreground elements in the scene. For each frame I_k, the sharpness was calculated using the LSPD method, giving:

$$U(\cdot) = LSPD(I_k) \tag{3.9}$$

3.8.2 Motion Indicator

Relative motion is a deciding factor in the expected performance of several components tested, and gives a general insight of scene behavior in each video evaluated. The motion of two successive frames will be determined by the single differencing motion estimator:

$$U(\cdot) = SingleDifferencing(I_k, I_{k-1}) \tag{3.10}$$

3.8.3 Background Model Initialization Amount

The TMSU and TMCSU background models will during initialization contain undefined (black) pixels of zero value. By measuring the amount of undefined pixels in the background estimates, one may get an indicate of how far the initialization process the background model has progressed. This indicator will yield inaccurate results when background scenes that naturally contain black pixels are estimated; however, the video clips analyzed seemed to rarely contain black pixels, making this a minor issue for the purpose of this thesis. For every pixel $p = (x, y)$, $U(\cdot)$ will be found from:

$$U(\cdot) = \begin{cases} 1 & , \quad \text{if } B_k^p = 0 \\ 0 & , \quad \text{if } B_k^p > 0 \end{cases} \tag{3.11}$$

3.8.4 Foreground Segmentation Amount

In order to document performance of the background segmentation procedure, it is of interest to know the percentile content of foreground pixels in the calculated binary foreground image. In his thesis, (Jakobsen, 2011) estimated that approximately 30% of a scene containing net structure would consist of foreground pixels, depending on the net thread diameter and the amount of algae growth. While finding a precise estimate for the amount of foreground pixels is difficult, in particular when binary images are combined and morphological operations are conducted, observing the amount of foreground pixels can be used as an indicator to how well the background segmentation process as a whole performs. Moreover, this information can, for instance, be used to adjust thresholding parameter for the DDT technique in various system components. Later in this thesis, it will be seen that the running mean of the sharpness indicator fairly accurately resembles the amount of foreground elements in the scene, which might be used as a relative reference for segmentation performance when compared to the foreground segmentation amount. For this indicator, $U(\cdot)$ will chosen as:

$$U(\cdot) = F_k \tag{3.12}$$

where F_k is the binary foreground image at frame index k.

3.9 Combining Image Channels and Segmentation Methods: The Unique and Uniform Design Schemes

It was concluded by (Kameda & Minoh, 1996) that employing a combination of color information from multiple channels, as well as edge and motion information, typically would improve performance and robustness in static background segmentation system. Since the primary objective of the background segmentation system of this thesis is to robustly and accurately segment an unpredictable, changing set foreground elements in a challenging environment from an presumingly static background, implementing robustness through redundancy seems like a reasonable approach.

The collaboration of image channels and segmentation methods presented in this section will be based on the two following arguments, which have been formed from the initial analysis and previous work from Chapter 1, as well as experiments from the subsequent chapters:

1. It seems unlikely that a single image channel comprehensively can distinguish foreground objects of different color, brightness, texture and behavior singlehandedly in a predictable and consistent manner; however, by combining the segmentation results from multiple image channels it might be possible to distinguish all foreground elements, given that the information of these foreground objects is contained within the analyzed multi-channel image.
2. Resulting from the various image characteristics utilized during calculation of motion, edge and temporal background segmentation, each segmentation method will naturally have its own set of advantages and deficiencies. In order to best isolate both known and unknown foreground objects, combining the segmentation results from methods with diverse properties seems like a rational approach to ensure robust operation in a practical implementations.

Table 3.5: Guideline arguments for a combinatorial system approach

Two multi-channel, multi-method designs of the “Background Segmentation Process” module in the main video analysis workflow in Figure 3.1 will be presented in the subsequent sections, namely the *unique* and *uniform* combinatorial component and parameter design schemes. The unique and uniform design schemes can be distinguished by their approach for parameter tuning, as well as their reliance in the performance of individual image channels and segmentation methods, reflected by the combinatorial layout of the components in each design. A compact summary and comparison of the two design schemes and their methodology has been stated in Table 3.7, while their system designs can be found in Figure 3.14 and 3.15 for the unique and uniform combinatorial background segmentation modules, respectively.

Unique vs. uniform combinatorial methodology:

In the unique combination design, all segmentation modules are maintained and calculated in isolated environments divided by each image channel. The binary foreground from each individual process is then combined. In the unique design, the comprehensive performance of each isolated process is crucial.

The uniform combination design employs an inverse methodology, where multiple image channels are evaluated and combined within each respective segmentation module. Consequently, the outside system does not see the performance of each individual image channel: it only sees the combined binary result from the segmentation module. In the uniform design, only the combined performance of all image channels is of importance, regardless of their individual behavior.

Unique vs. uniform parameter schemes:

The unique parameter scheme employs a different set of parameters for every clip studied, where each set is specifically tuned for each particular clip analyzed. The parameter selection aim to provide the most comprehensive segmentation result for each independent image channel and scene.

The uniform parameter scheme aims to find a single set of parameters for all modules it incorporates that gives a stable and predictable behavior when applied to all clips analyzed, a well as scenes not specifically prepared for. It is assumed that by using suboptimal, but robust, parameter settings, the most distinct foreground objects in each image channel will in total produce a comprehensive combined segmentation result.

Table 3.7: Summary and comparison of methodologies for the unique and universal design schemes.

By combining multiple image channels and segmentation methods, the computational complexity combined system will be increased. However, since only real-time applicable algorithms are used, and the computational demands of including multiple channels scales linearly, this will most likely not be a practical concern for the unique and uniform designs.

3.9.1 Unique vs. Uniform Design Scheme: A Case study

A comparison of the unique and uniform design schemes for segmenting foreground objects with some method incorporating binarization through DDT, such as the LSPD, TMCSU and single differencing submodules in Figure 3.14 and 3.15, will now be given. The case will investigate the scene in Figure B.1c, which is typical scene in the C1 video clip. As will be seen in Section 4.1, the blue color channel is particularly effective for distinguishing fish and algae growth the background, whereas the green channel is better for distinguishing net structure.

Only isolating the net structure from the background with the green channel using DDT binarization is typically no big concern, since the net wall is well distinguished

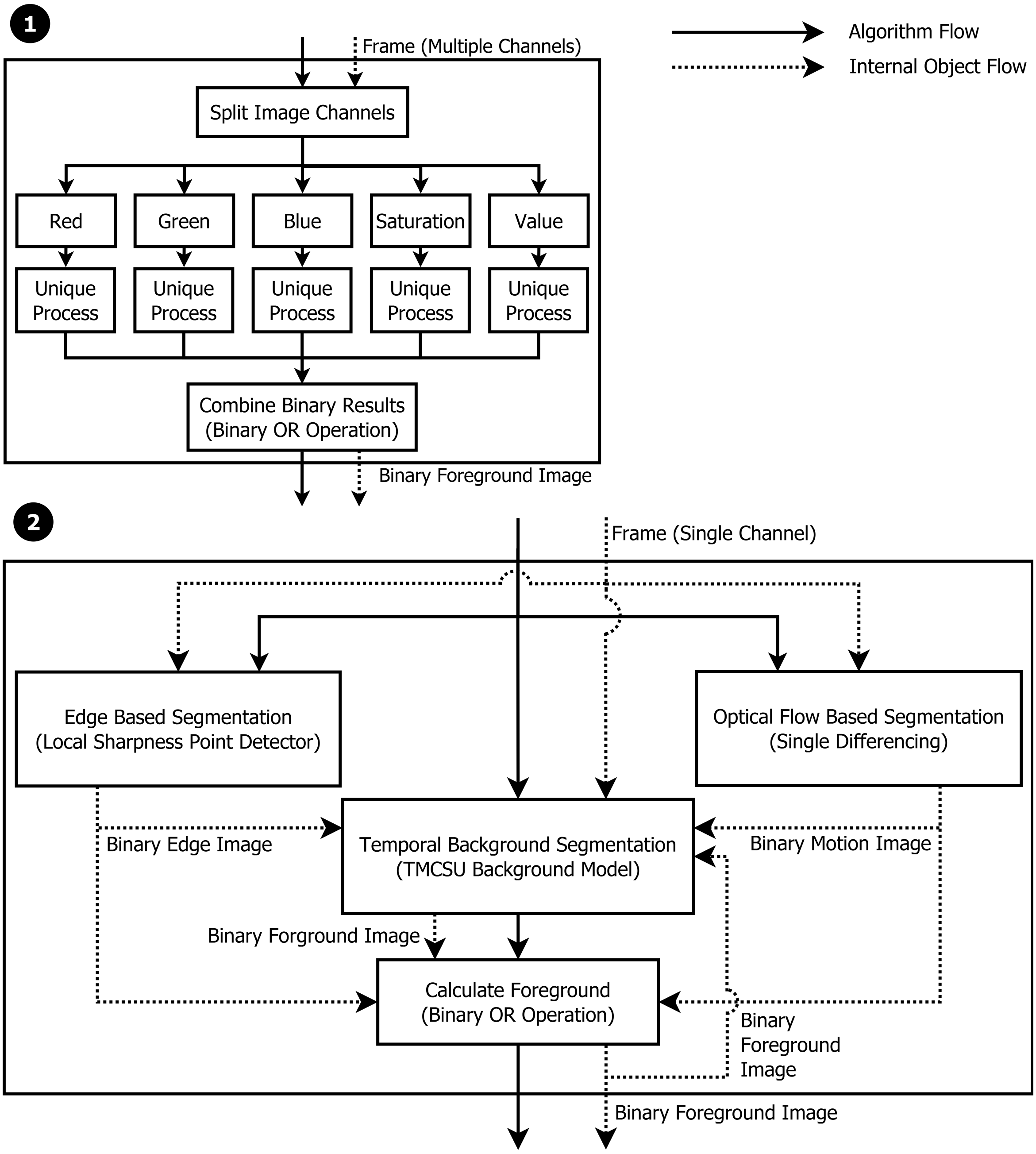


Figure 3.14: A unique, combinatorial system design: **1.** Unique background segmentation process main module, **2.** Unique process submodule

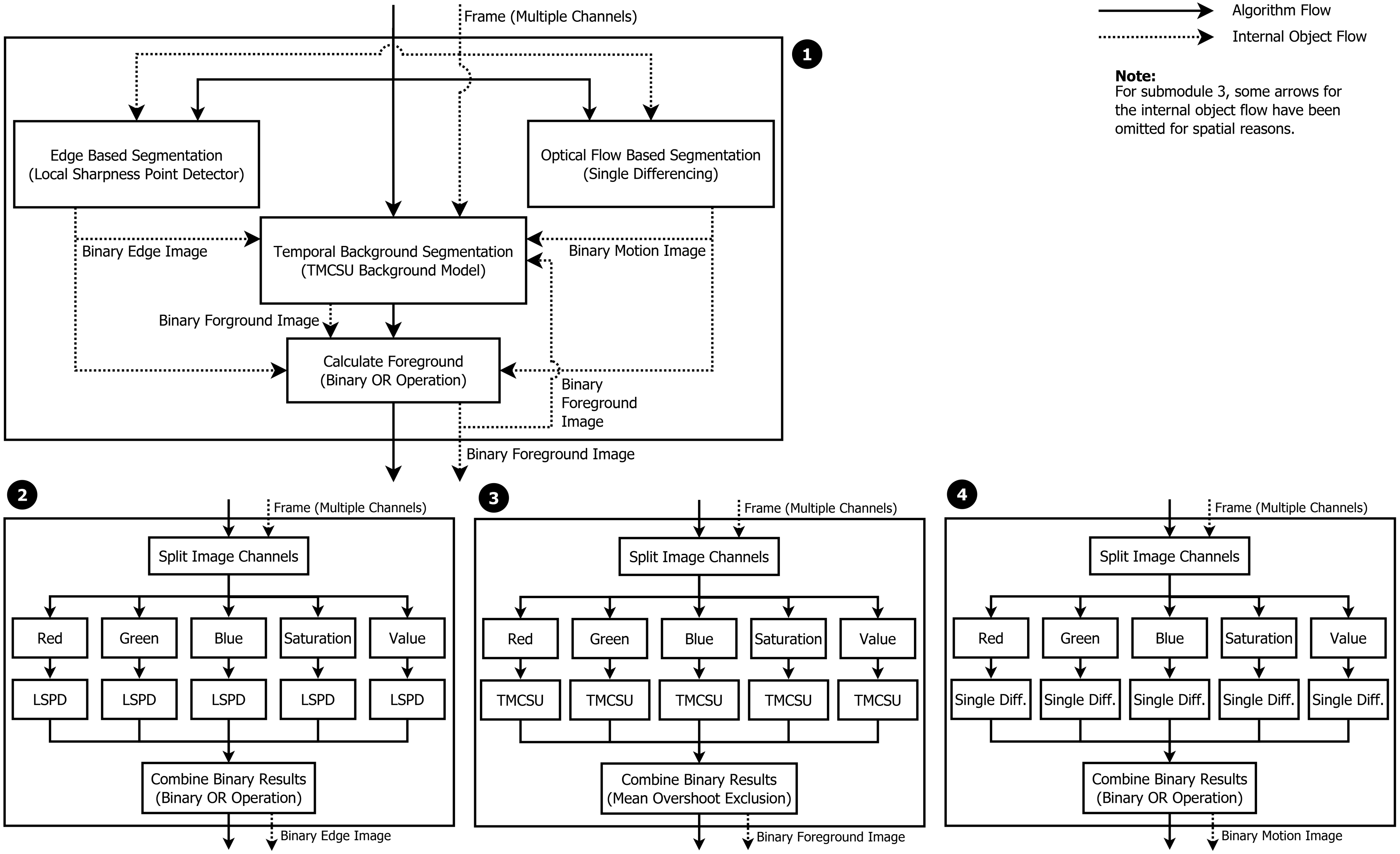

Algorithm Flow
Internal Object Flow
Note:
For submodule 3, some arrows for the internal object flow have been omitted for spatial reasons.
1
Frame (Multiple Channels)
Edge Based Segmentation (Local Sharpness Point Detector)
Optical Flow Based Segmentation (Single Differencing)
Temporal Background Segmentation (TMCSU Background Model)
Binary Edge Image
Binary Motion Image
Binary Forground Image
Calculate Foreground (Binary OR Operation)
Binary Foreground Image
Binary Foreground Image
2
Frame (Multiple Channels)
Split Image Channels
Red
Green
Blue
Saturation
Value
LSPD
LSPD
LSPD
LSPD
LSPD
Combine Binary Results (Binary OR Operation)
Binary Edge Image
3
Frame (Multiple Channels)
Split Image Channels
Red
Green
Blue
Saturation
Value
TMCSU
TMCSU
TMCSU
TMCSU
TMCSU
Combine Binary Results (Mean Overshoot Exclusion)
Binary Foreground Image
4
Frame (Multiple Channels)
Split Image Channels
Red
Green
Blue
Saturation
Value
Single Diff.
Single Diff.
Single Diff.
Single Diff.
Single Diff.
Combine Binary Results (Binary OR Operation)
Binary Motion Image

Figure 3.15: A uniform, combinatorial system design: **1.** Uniform background segmentation process main module, **2.** Uniform LSPD submodule, **3.** Uniform TMCSU submodule, **4.** Uniform single differencing submodule

from the background. As a result, the static DDT thresholds can be set to moderate levels, where no immediate change in the segmentation of the net structure would occur if the thresholds were slightly adjusted, or the scene underwent moderate changes. Equally, the blue channel would most likely manage to segment the algae growth and the fish passing through the scene very well without straining the thresholding values in either direction, allowing the scene to undergo moderate changes without over- or undersegmenting the image, while still providing a satisfactory segmentation of the fish and algae growth. Intuitively, combining the binary segmentations from the green blue image channels would provide a combined foreground image containing most, if not all, foreground objects in the scene, while allowing moderate changes in the scene without altering the combined result. Due to the natively robust nature of this segmentation process, modules in the parent system could utilize the combined binary result while ensuring optimal operation. This manner of combining image information is the methodology of the uniform parameter scheme, and the approach utilized within the LSPD, TMCSU and single differencing submodules of the uniform combinatorial design in Figure 3.15.

In the unique parameter scheme, the aim of the green image channel would be to segment the net structure, fish and algae growth in one single operation. Isolating the net structure would most likely be trivial, since it is well definable from the background scene; however, segmenting the algae growth with the green channel poses more of a challenge. Since the algae growth barely distinguishes itself from background scene, very low binarization thresholds have to be used in order properly segment the algae growth. Due to these very low thresholds, even the slightest change in the scene would potentially result in an erroneous segmentation of background pixels, meaning that the thresholds selected only would function properly in the very specific scene conditions they were selected for. Furthermore, since another module depends on a comprehensive segmentation result from the green channel, selecting moderate values, as for the uniform parameter scheme, is out of question. The blue channel is opposed with the a similar dilemma to that of the green channel. However as it turns out, a shadow is falling on a region of the net, and for the blue channel, the net structure in this region is barely distinguishable from the background scene due to the equal brightness levels. As a result, it is impossible to find a set of threshold values that manages to segment the entire net structure without also erroneously segmenting background pixels. A compromise in setting the threshold values was therefore made, so that most, but not all, of the foreground was included in the segmentation result, including some background pixels. Small changes in the scene would most likely degrade the blue segmentation result, and the other modules that depend on this result would have to work with an overall suboptimal binary foreground segmentation. At some later point in the parent system, the binary segmentations from the green and blue image channels are combined, ultimately resulting in a binary foreground image with some erroneous segmentation, and possibly also oversegmentation, since both binary images contain slightly different segmentations of the same foreground elements. In summary, both image channels would could potentially generate a segmentation fragile to scene changes, while providing suboptimal binary results to components in the parent system and possibly yield an erroneous combined segmentation result for further usage. These are among the challenges of the unique parameter scheme and combinatorial design given in Figure 3.14.

3.9.2 Comparison, Further Usage and Test Routine

The case analysis in the previous section was derived from general observations made during investigating of the TMBU, LSPD, single differencing and temporal background subtraction which will be documented in some format in the next chapter, and also highlighted several points worthy of further discussion.

In several ways, the isolated channel processes of the unique design scheme resembles the design criteria and behavior of a typical static background segmentation system that does not rely on information from multiple image channels nor interconnected segmentation methods. And when such a design is set to a multi-channel, multi-segmentation context, as in the combinatorial unique design of Figure 3.14, several features that does not adapt very well to a combinatorial design mishap. In general, there is a conflict between the performance of the isolated process of individual channels and the aim of the combined system: improving the segmentation coverage of the of the first will easily degrade the performance of the latter, a degrading effect that upscales with the inclusion of further image channels. In addition, neither the combined nor isolated processes of the unique combinatorial design scheme employs a strategy of achieving robustness to changing scene conditions with a non-adaptive binarization method, such as DDT. And for these reasons the unique design scheme will not be utilized in the final system design described in Section 3.11. It will, however, be used a counterpart to the uniform design scheme for testing purposes, as it highlights the individual potential of each segmentation method and image channel very well, which is, the segmentation results that could have been if an automatic thresholding scheme was utilized.

Unlike the unique design scheme, the uniform combinatorial design and parameter scheme in Figure 3.15 has been developed specifically for a modular multi-channel, multi-segmentation collaboration, where the strengths of each component work towards a common goal while keeping all thresholding parameters within acceptable margins. With the binarization method investigated in this thesis, the assumption that combining only the strongest, most distinct segmentation points from each image channel - and segmentation method - in a complementary fashion, might provide a desirable system robustness despite utilizing static threshold values. If, indeed, this assumption holds, the uniform combinatorial design and parameter scheme will most likely prove favorable over the unique design scheme, while allowing for a reliable final segmentation result with a high level of redundancy to manage foreground objects of unknown character, as well as temporal and overall changes in the scene. For these reasons the uniform design scheme in Figure 3.15 will be utilized in the final system, in combination with the LSPD, TMCSU and single differencing methods. As will be seen later, each of these segmentation modules excel at segmenting various foreground elements, and therefore largely complement each other.

Due to the complexity of the module designs in Figure 3.14 and 3.15 for the unique and uniform schemes, respectively, the experimental comparison of the two will not include complete tests of both systems. Instead, the performance of each image channel of the segmentation methods incorporated will be studied in an individual and combined fashion, in order to anticipate their potential usage, flaws and strengths. Specifically, all tests will utilize the generic test module illustrated in Figure 3.16 with a unique and uniform parameter scheme, from which their performance will be eval-

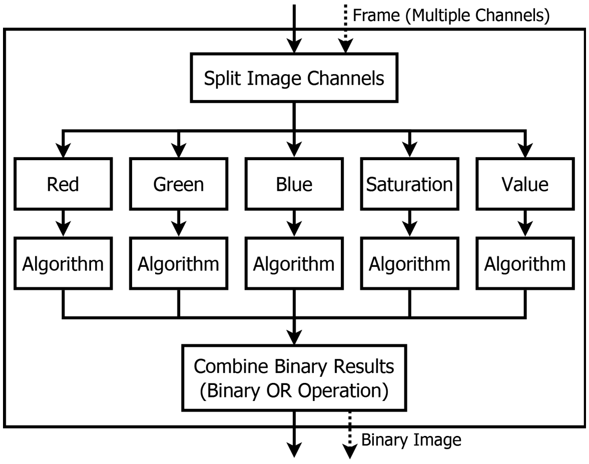


Figure 3.16: Generic test module for comparing unique and uniform component performance

uated according to the methodology of each combinatorial design scheme. The hue channel will not evaluated for either design, due to an overall poor response, as will be seen in Section 4.1.

3.9.3 Overshoot Exclusion

Contrary to the unique design scheme, the modular character of the uniform design in Figure 3.15 allows for the incorporation of different binary additive operations for each segmentation module. When binary images are combined using OR operations, oversegmentation in the resulting binary image can potentially occur. If each module behaves well for various situations, then using a binary OR operation for combining the segmentation results can be considered to be of low risk. However, if one or several segmentation modules occasionally tend to misbehave by oversegmentation, then the final combined binary image would occasionally also be degenerated.

It was mentioned in Section 3.7 that an outdated background estimate most likely would result in an oversegmented binary image from the temporal background segmentation methods. As an illustrative manner of managing such potential issues, a simple locking mechanism will be utilized when combining the binary results from the TMCSU model in the uniform design scheme. The method, which will be referred to as *overshoot exclusion*, functions as an ordinary binary OR combinator which effectively excludes any input image from the OR operation if their percentile content of foreground pixels is larger than some predefined limit, L, as visualized in Figure 3.17. With an appropriately set limit, the oversegmentation from swift scene transitions in C3, and similar cases, might be handled.

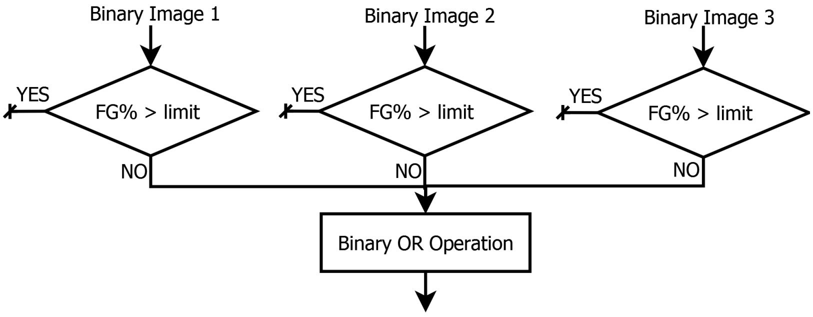


Figure 3.17: Overshoot exclusion for three binary images

3.10 Damage Detection

The final evaluational step of the net damage assessment system developed in this thesis, outlined in Figure 3.1, is the operation of damage detection. Due to the unpredictability of the scenes analyzed and foreground objects encountered, the foreground structures, such as the net threads, in the background segmentation results cannot always be assumed intact. The segmentation results will therefore be analyzed using a morphological closing operation where the structuring element can be adjusted to identify damages of a user-defined size and shape. Contrary to detecting damage with region growing, as purposed by (Sletta, 2013), this approach will not require the foreground structures in the binary foreground image to be perfectly intact.

A net damage is in this thesis defined as a sizable image region not covered by some sort of foreground element, as described in Section 1.3. Since the background segmentation process provides a binary foreground image, a closing operation will be used in order to fill all regions less than some specified size, where unfilled regions will be classified as net damage. The size, shape and orientation of the damages to be detected can be controlled by the size, shape and orientation of the structuring element used in the closing operation, where only background regions able to completely enfold the structuring element are left unfilled. It should be noted, however, that a background region only completely enfolds a structuring element if no foreground pixels, including noise and isolated spots, are present in that region. As such, the physical size of the structuring element should usually not be matched to the exact size of the damages to be detected. Furthermore, the distance between the camera and the net naturally affects which damages are detected when a structuring element of predefined size is used. In a practical implementation, this should be accounted for by adjusting the size of the structuring element to the distance to the net.

In Figure 3.18, the damage detection module that will be utilized in the final system is illustrated. In addition to the morphological closing operation, both a median filter and an opening operation is present. The median filter has been included in order to remove initial foreground noise pixels from the binary image before the closing operation. The opening operation, on the other hand, has been incorporated to bring background regions erroneously shirked by the influence of isolated spots of foreground and noise back to the original size of the damaged region. In order to avoid removing vital image detail, the median filter will use a template of small size, while the opening operation will use the same structuring element as the closing operation to not expand the region of the damage beyond its original size. Furthermore, a basic

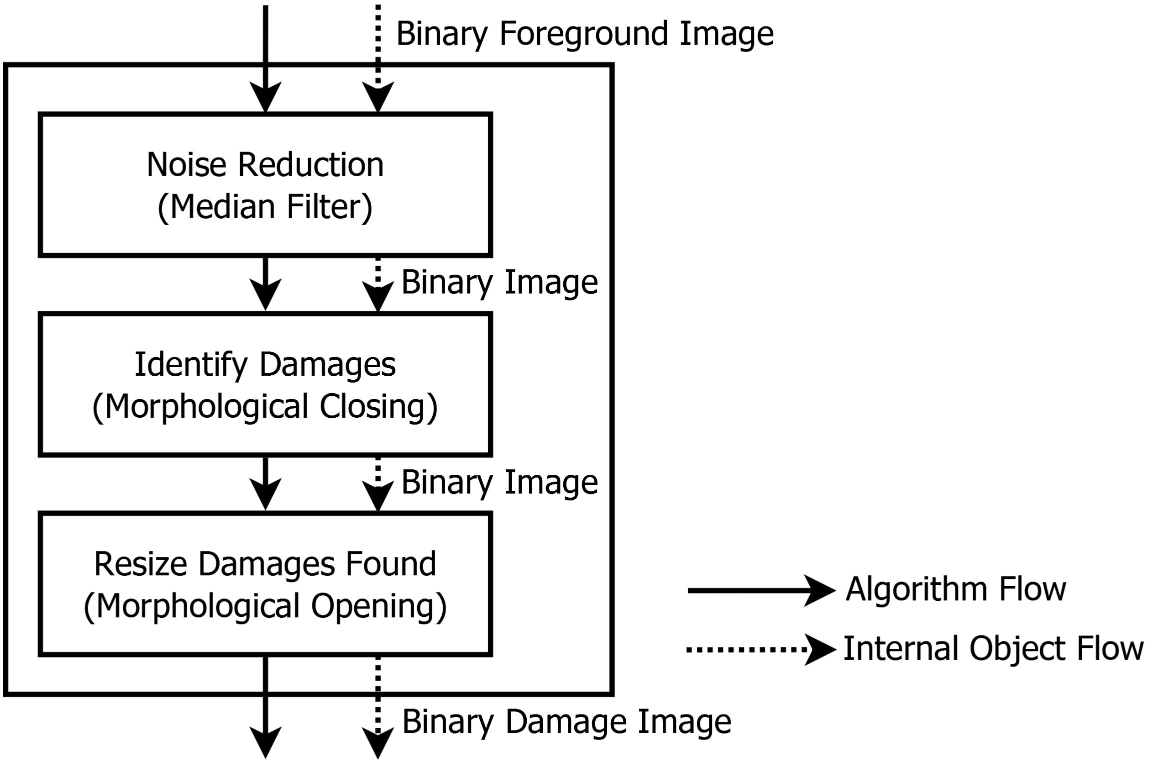

Binary Foreground Image
Noise Reduction (Median Filter)
Binary Image
Identify Damages (Morphological Closing)
Binary Image
Resize Damages Found (Morphological Opening)
Binary Damage Image
Algorithm Flow
Internal Object Flow

Figure 3.18: Damage Detection Module

disc-shaped structuring element will be utilized for the morphological operations, due to its orientation-independent nature. The radius of the disc will therefore be the user-definable parameter for deciding the size of the net damages to be detected.

The damage detection algorithm in Figure 3.18 will be evaluated along with the final system, but only on the net damage simulations contained in the C3 video material; after all neither C1 nor C2 contains any knowledgeable actual net damage. The tests will attempt to find a functioning radius of the structuring element, and a suitable template size for the median filter. Since no measure of net distance is made in this thesis, an automatically scaled structuring element will not be explored.

3.10.1 Usage of Binary Detection Result

The final binary image resulting from the damage detection algorithm will contain connected regions of either black or white pixels, representing net damage and foreground structure, respectively. Due to this unambiguous distinction of damaged regions in the resulting binary image, further utilizations of the damage detection results can be made equally simple.

In Section 1.1, marking damage detected in live or recorded video streams for assisting an ROV-operator to identify net damages was suggested. This task could be accomplished by overlaying a color mask or region contour of the binary detection image in video, or similar.

In a fully autonomous detection system, the center position of the damaged regions might be of particular interest, since these could serve as a navigational references for the AUV's path planning algorithm. Considering the simple nature of the binary detection image, such a position could, for instance, be calculated from the average (x, y)-position of black image pixels, in either: the entire binary image, for situations where only a single damaged region is detected; or within each region, in the case where a single image contains multiple detected damages. Blob detection could be another alternative.

3.11 Final System Design

In this chapter, a multitude of algorithms and component designs have been suggested, described and illustrated. Several of these methods are forerunners or submodules of more advanced systems intended to make the most complex combinatorial modules comprehensible, both in terms of parameter selection, as well as in understanding internal system functionality and behavior. In the next chapter, the individual tests of these components will be reviewed, all results of which leads to the system that will be summarized in this section, that is: the final system.

The final system design will be based on the video analysis outline illustrated in Figure 3.1, where the content of each module will be as listed in Table 3.8:

Design of the Final System (based on Figure 3.1):

Noise Reduction

- Gaussian Filter, smoothing filter

Background Segmentation Process

- Uniform Combinatorial Design (Figure 3.15)
    - LSPD, edge detector (Figure 3.8)
    - Single Differencing, motion estimator (Figure 3.6 (Unit 1))
    - TMCSU, temporal background segmentation (Figure 3.12)
        - S&KB-TMCSU, update scheme (Figure 3.13 (Unit 2))
- Overshoot Exclusion, binary image combination (Figure 3.17)

Damage Assessment

- Damage Detection Module (Figure 3.18)

Table 3.8: Composition of the final system of this thesis

Due to the intricacy of the final system, tests conducted will not focus on the internal behavior of individual components, but rather the compatibility and cooperation of the results calculated from the main modules, which are the uniform the LSPD, single differencing and TMCSU segmentation modules. The combined performance of these modules in terms of their ability to robustly provide an accurate and consistent binary foreground segmentation will be studied, a result of which we used as input for the damage detection algorithm.

Chapter 4

Experimental Results: Components

In this chapter, the properties of individual algorithms and system components will be tested, analyzed and evaluated. The aim will be to acquire knowledge about the strengths and weaknesses of each component, and determine their practical usability in the design of the final system, which was presented in its concluding form in Section 3.11. Several of the components analyzed are included in an attempt to determining the exact, isolated effects of adjusting specific parameter settings. This is motivated by the somewhat overwhelming amount of tuning parameters in the final system (Table 3.8), where the outcome of modifying individual parameters easily may be obscured by the complexity of the system.

Due to the fundamental nature of the tests reviewed in this chapter, some readers might want to skip to Chapter 5, where the results of the final system developed in this thesis are reviewed. However, if the reader is interested in the design decisions of the final system, the individual behavior of internal components, how parameters are selected etc., all these experiments, result analyses, discussions and conclusions are contained within the subsequent sections of this chapter. A such, a modular structure has been chosen for this chapter in order to allow the reader to look back at individual topics of interest, without having to reading all previous sections.

In Figure 4.1, a stepwise overview of all tests conducted in this chapter has been illustrated. The steps 1 to 11 describe the relative time at which the tests were performed, while the arrows display which of the previous tests for each module experiment depends on; for instance, the LSPD method only utilizes noise filtered input from the ”best” color space, while the “Combine Method & Image Channels” module depends on the LSPD module both directly and indirectly through the TMCSU method. An exception to the time perspective of this overview, is the frame quality indicators, which will be used for analytical purposes during data collection and discussion of several experiments presented in this chapter. Also note that the modules marked with a $\star$ in Figure 4.1 will be tested in combination with the final system in Chapter 5, and not in this chapter.

All experiments conducted will follow the generic video analysis system outline in Figure 3.1 in a top-down manner; for instance, when the “Background Segmentation Process” block of this workflow is evaluated, the error correction and noise filtering steps prior to this block will be assumed always active. Exactly which algorithm each block will incorporate in previously tested steps will be considered in each block's respective section of this chapter.

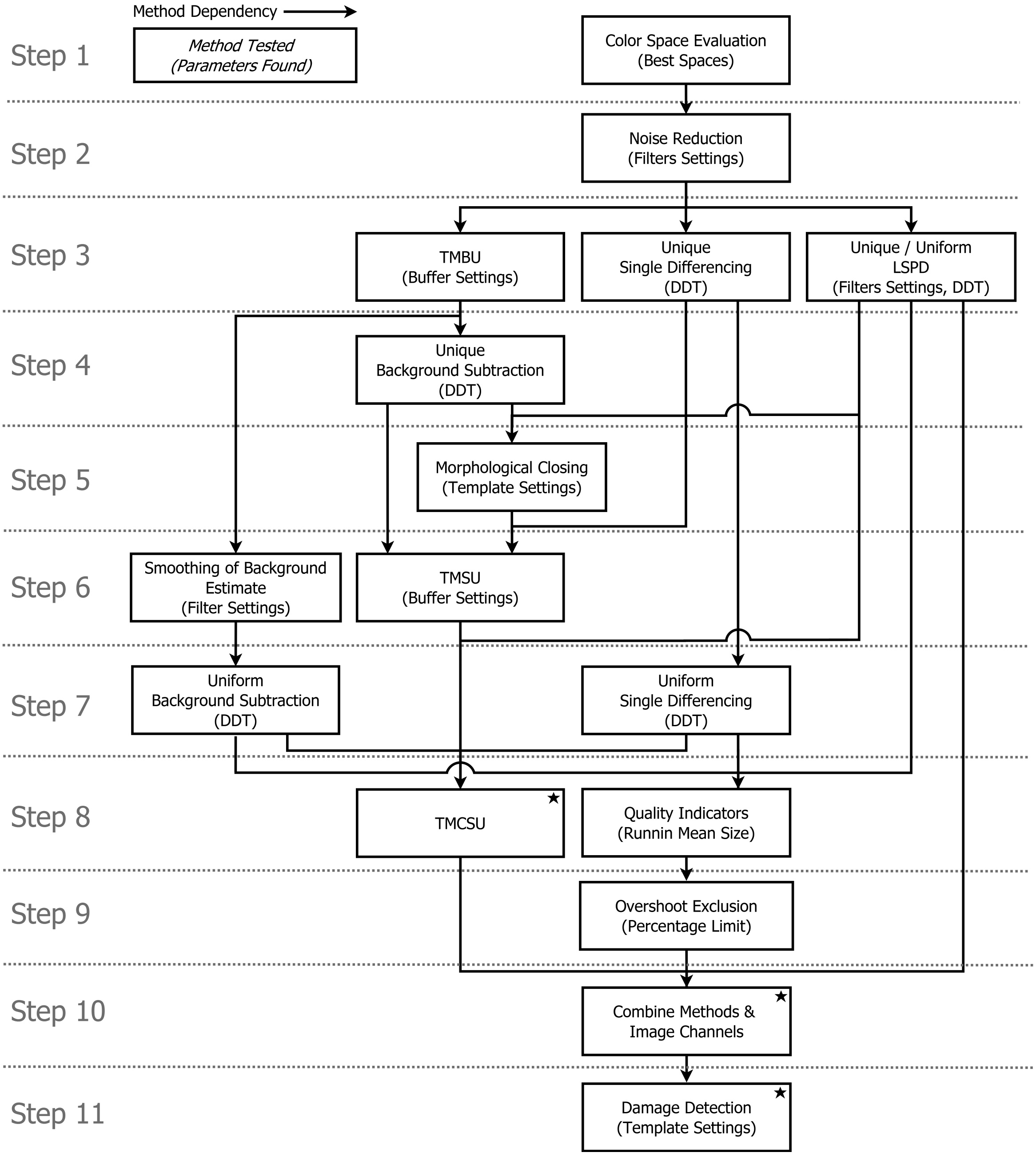


Figure 4.1: Stepwise order of component tests. Tests marked with a ⋆ will be conducted in Chapter 5

4.1 Color Space Evaluation

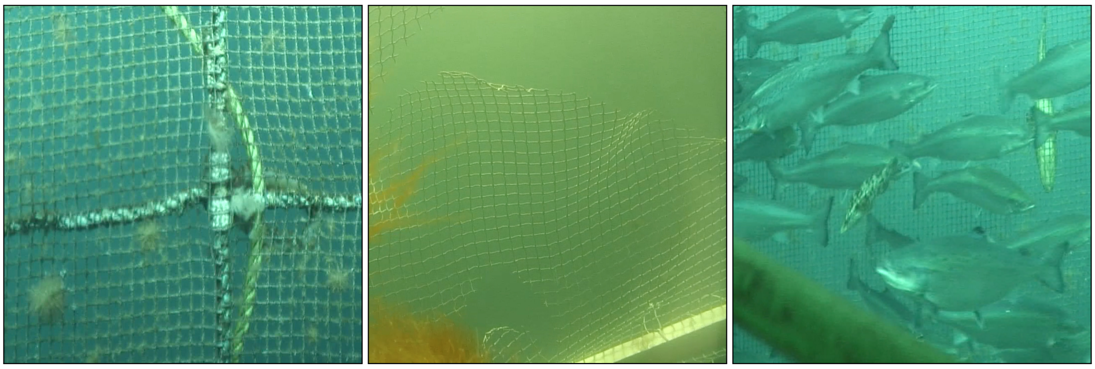
It is stated by Plataniotis that, regarding the appropriate choice of an image's color model:

> A well chosen representation preserves essential information and provides insight to the visual operation needed. Thus, the selected color model should be well suited to address the problem's statement and solution (Plataniotis & Venetsanopoulos, 2000).

As introduced in Section 1.5, and later discussed in Section 3.9, a single color channel will most likely not be able to preserve the information essential for yielding a complete, continuous background segmentation result. The color model most suited for the application of this thesis might therefore benefit from combining properties from multiple image channels. In this section, properties of the RGB color spaces and the HSI family of color spaces, both of which are commonly employed in computer vision, will therefore be analyzed, compared and evaluated.

4.1.1 Test Setup and Evaluation Criteria

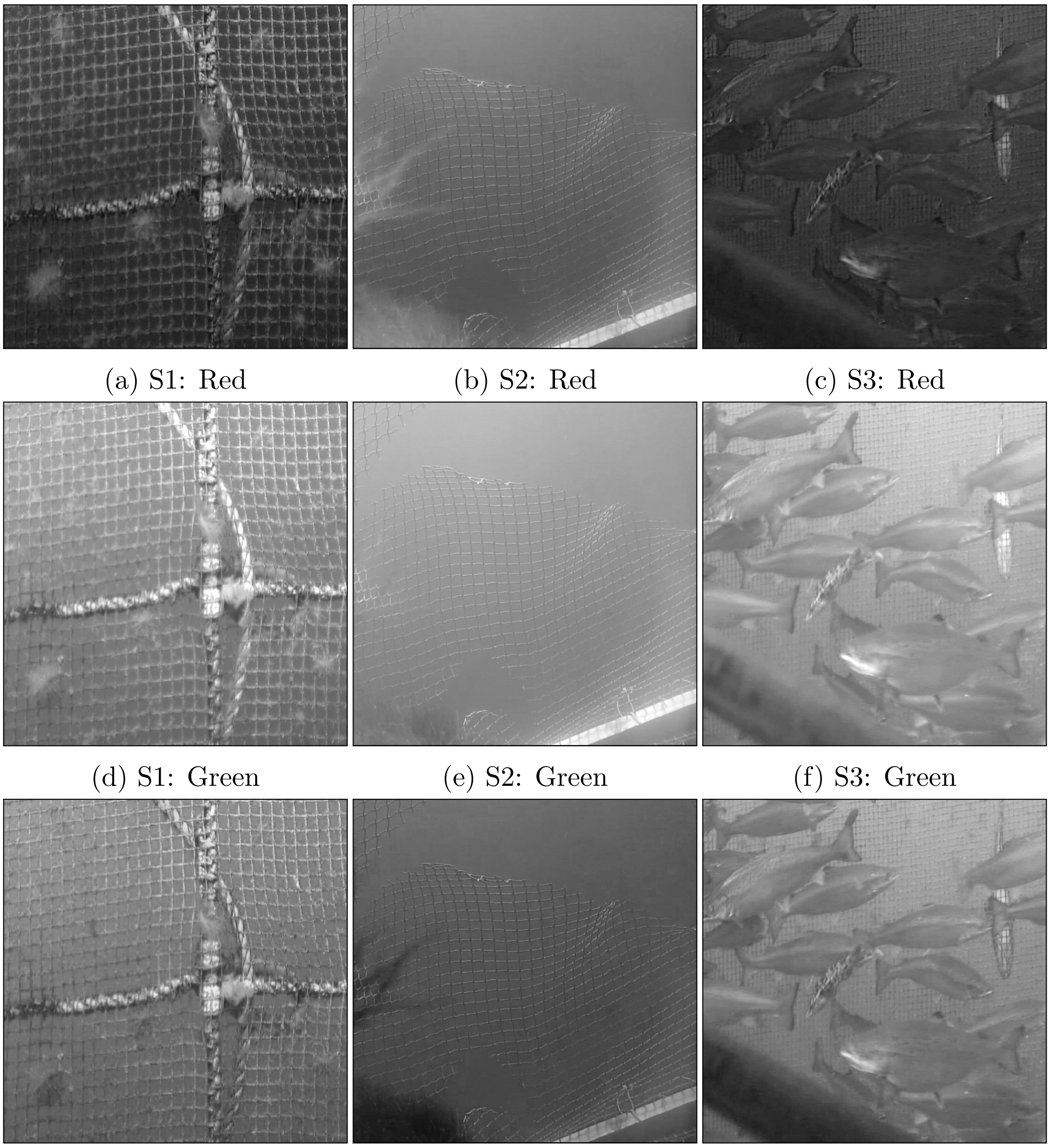
In Figure 4.2, three test image that will be used for evaluating each color channel is displayed. These test images have been selected to contain as many of the challenges discussed in Section 1.5 as possible in order to fully assess the separation capabilities of each respective color channel for typical image elements. The primary objective will be to find a combination of color channels that contain the information needed to completely isolate all foreground elements from the background. Since the background is fairly homogeneous, such image information will not blend with the background, meaning it will have a different tone of gray in the displayed examples in this section. Each channel's ability to uniquely and robustly describe identical objects in different scenes and conditions, here referred to as their *information format consistency*, is also of importance, as a high format consistency could simplify the post processing, such as selecting binarization parameters. The evaluation of each color space will not be based only on the test images in Figure 4.2, but also on their respective video material; however, these images manage to display most of the foreground objects contained in the video material in a decent manner.



(a) Scene 1 (S1) (b) Scene 2 (S2) (c) Scene 3 (S3)

Figure 4.2: Original images for color space comparison

The test images in Figure 4.2, are displayed for the RGB channels in Figure 4.3, while, a comparison of the HSI, HLS and HSV color spaces - that is, their the intensity, lightness and value channels - is shown in Figure 4.4. From this comparison, the overall brightness of each channel seems to be the only distinguishable difference between the three color spaces. Due to its widespread usage in computer vision and available software support, the HSV color space will therefore be used to represent the HSI family of color spaces in this thesis. Consequently, each test scene has been investigated for the complete HSV color space only, as visible in Figure 4.5.



(a) S1: Red

(b) S2: Red

(c) S3: Red

(d) S1: Green

(e) S2: Green

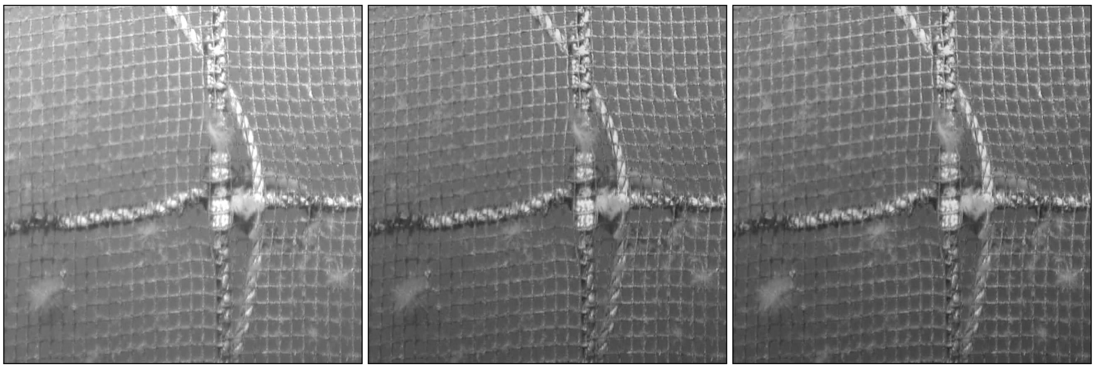
(f) S3: Green

(g) S1: Blue

(h) S2: Blue

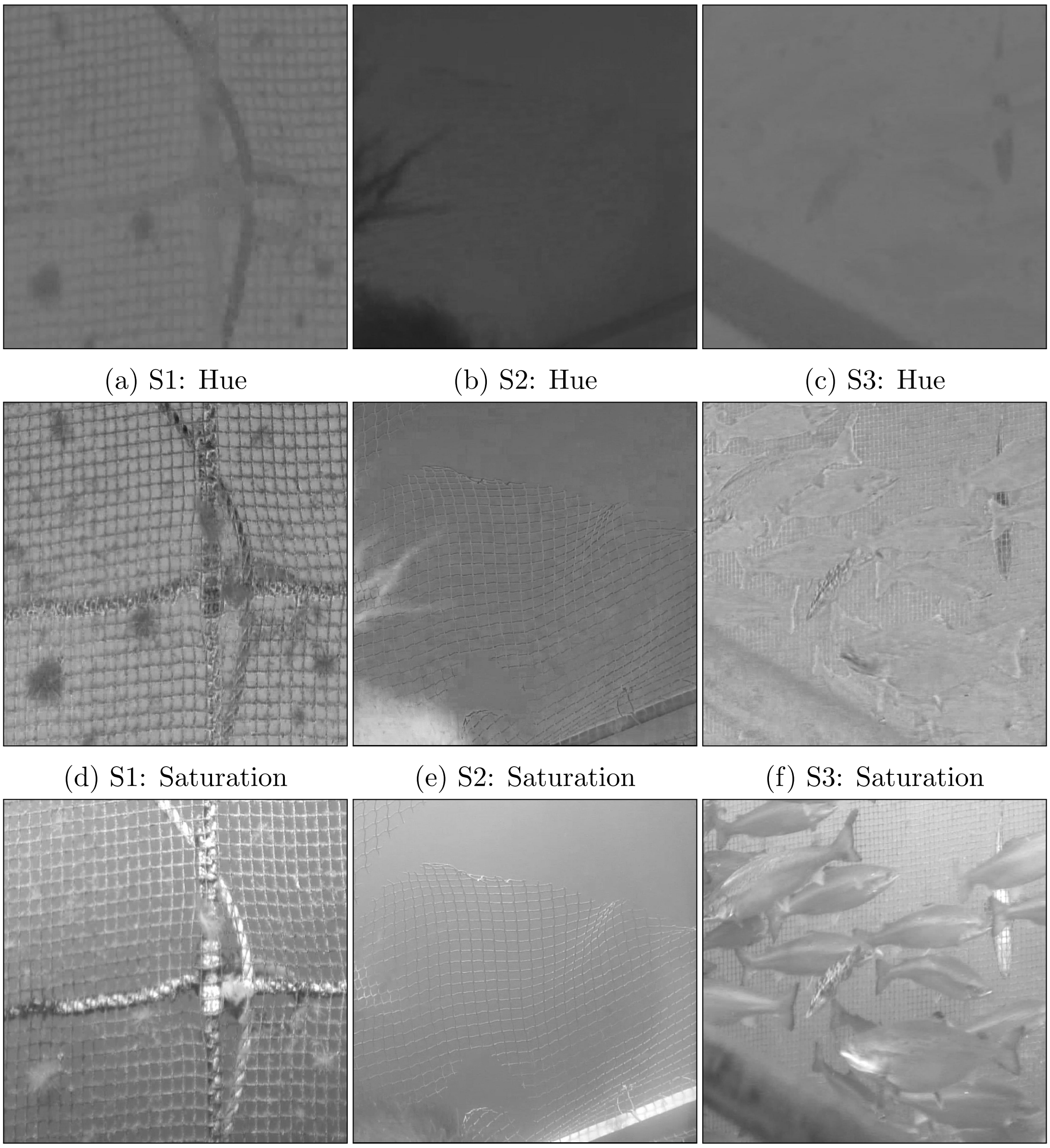
(i) S3: Blue

Figure 4.3: RGB image channels



(a) S1: Value (b) S1: Intensity (c) S1: Lightness

Figure 4.4: Comparison between the HSI, HSV and HLS color spaces



(a) S1: Hue (b) S2: Hue (c) S3: Hue

(d) S1: Saturation (e) S2: Saturation (f) S3: Saturation

(g) S1: Value (h) S2: Value (i) S3: Value

Figure 4.5: HSV image channels

4.1.2 Analysis of Test Results

The following observations were made during testing of each individual color channel:

- **Red**: The red channel was overall darker than the green and blue, with marked contrast in the net threads. In some cases, it was harder to separate fish from the background with the red channel, but in areas with dominant algae growth overlay, the channel had a slightly better separability than the green channel.

- **Green**: The green channel seemed to reproduce foreground elements and complete net threads slightly more consistently than the red and blue channels. Also, in the samples tested, the green channel had a fairly constant brightness level. The artificial growth in Figure 4.3e, however, is barely distinguishable.

- **Blue**: The blue channel seemed to markedly separate growth and tether cable, and distinguished fish from the background very well, but with a relatively poor Signal to Noise Ration (SNR), as seen in Figure 4.3g and 4.3h.

- **Common for RGB**: Each RGB channel occasionally gave better or worse separability of the background than the others, subject to variations in lightning, the type of foreground object and the scene. In general, there were very few definite differences apart from an overall sensitivity to lightning and changes in the scenes: the same foreground objects would appear both brighter and darker than the background depending on the situation.

- **Hue**: The hue image channel showed no particular beneficial traits on any of the video samples or scenes tested. At close range certain elements were distinguishable from the background, in particular algae growth, and to some degree the net structure. In most other cases, however, the hue channel yielded no distinguishable information at all, and what was visible carried little detail and appeared blurred. In video feeds, the hue singularity was occasionally visible as entirely black or white pixels.

- **Saturation**: The saturation channel distinguished certain kinds of foreground elements - in particular net structure, growth and ropes - from the background very well, despite varying light conditions and shadows. On the other hand, other foreground elements, such as fish or tether cable, blended entirely with the background. When analyzing the video material from Scene 2 (Figure 4.5e), the saturation channel gave a somewhat chaotic output influenced by what appeared like discretization error - particularly in the background and the slightly blurred areas. Consequently, the saturation channel seemed to best separate the net structure at medium to close range, where noise had a less diminishing effect on image quality; at long range less detail could be seen (Figure 4.5f). Furthermore, the information format of the saturation channel was reversed between some scenes: in Scene 1 (Figure 4.5d), the foreground elements are mostly darker than the background, while the opposite is true for Scene 2 (Figure 4.5e).

- **Value**: The value channel managed to distinguish most foreground elements nicely, with results very similar to that of the green color channel, but with a slightly different response to chroma. In Scene 2 (Figure 4.5h), the chromatic limitation of this channel is visible by the low separability of the strongly colored artificial algae growth.

4.1.3 Summary

From analyzing each color channel individually, it seems like no single image channel managed to fully separate all of the featured foreground objects from a homogeneous background in a robust manner: each channel was able to distinguish different foreground objects under different conditions. However, in general, all of the image channels, except the hue channel, managed to differentiate large portions of the foreground elements from the background with very similar results in most situations.

The saturation and hue channels appeared to be relatively invariant to transient light conditions in each individual scene. In every other image channel, foreground elements were both darker and brighter than the background - depending on the light conditions. A low SNR limited the saturation channel for analysis at long range, while the hue channel barely responded even at close range. In practical terms, the range limitation of the saturation channel might not be an issue, as analyzing the net from a long range is undesirable in general, as discussed in Section 1.5. The hue channel, however, despite showing a consistent information format, will not be used due to its poor ability to reproduce detail even at close range.

Among the RGB channels, the blue channel showed the better differentiability for foreign objects and fish, while a poor SNR reduced its capability to distinguish individual threads in the net structure. The green channel had a very good SNR for all scenes, which made it excellent for seeing individual net threads in areas without algae growth. In terms of seeing the net structure through a layer of algae growth, the red and blue color channels performed better.

4.1.4 Discussion, Conclusion and Further Usage

Since no single image channel managed to distinguish all foreground elements equally well, a combination of image channels will be used in the final system of this thesis. Specifically, the red, green, blue, saturation and value channels - RGB & SV- will be utilized further, while the hue channel will be excluded. Whether all five, or just a selection of these, color channels truly are required in order to fully distinguish all possible foreground objects from some background scene, will not be determined in this thesis. Presumably, the RGB channels alone would have been sufficient for this purpose; however, by including the value and saturation channels, which incorporates a different set of properties and response characteristics and thus also different parameter selections for binarization etc., the additional redundancy might improve the overall system robustness to changing scene conditions and foreign objects not analyzed in this thesis.

The exclusion of the hue channel is unfortunate, as it potentially could have been a valuable resource for improving accuracy and robustness of the background segmentation procedure (Karaman et al., 2005), in particular for distinguishing strongly colored objects, such as algae growth, the tether cable and similar. The consistent information format of the hue channel could potentially allow for a highly robust foreground segmentation by directly employing basic thresholding techniques, despite changing scene conditions. Adversely, based on the video material analyzed in this thesis, the hue channel does not seem suited for further usage.

Visual experimental results documented in this and the next chapter will typically be displayed in the red or blue color channels, as they have shown an overall decent ability to distinguish various scenes and elements, while they also complement each other in several aspects, as will be seen later.

4.2 Noise Reduction

In an attempt to reduce the discretization noise in Section 4.1 found to diminish image details in some the image channels (the blue and saturation channels in particular), the effect of employing a smoothing filter to all input frames was briefly investigated in Section 3.3, with the following conclusion: by applying a Gaussian smoothing filter with standard deviation $\sigma = 3$ and template size $T_s = 3$, the discretization noise was largely suppressed while most details were preserved. The smoothing effect from the Gaussian filter is believed to slightly improve the consistency of the background estimate and subtraction result, overall improving segmentation performance. The remaining component and system tests of this thesis will therefore incorporate this initial smoothing operation.

4.3 Local Sharpness Point Detector

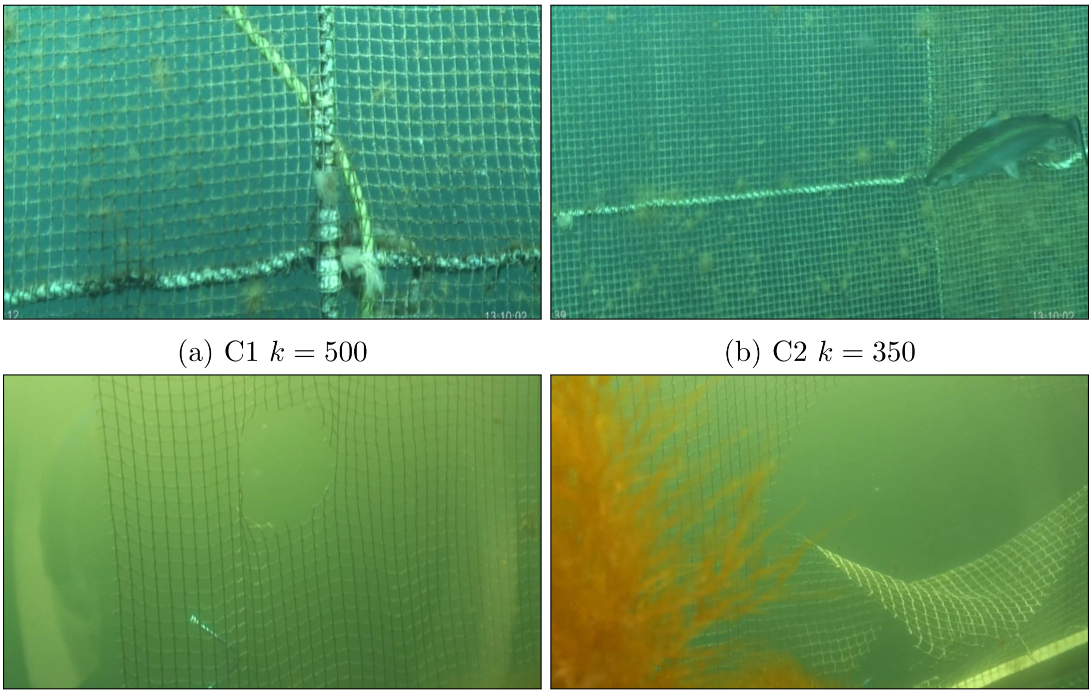
The LSPD algorithm was proposed in Section 3.6 as a basic alternative to traditional edge detector algorithms designed to provide a single edge response with good localization. In this section, these properties will be investigated in addition to the noise management and smooth object detection capability of the algorithm. Furthermore, the algorithm will be evaluated in terms of both the unique and uniform combinatorial system designs in 3.9.

4.3.1 Test Setup, Evaluation Criteria and Analysis

The aim of the LSPD algorithm will be to robustly detect sharp image objects, that is, independent of scene conditions, without providing an over- or undersegmented binary result that potentially could conflict with other modules when used in a complex system. In Figure 4.6, four images from the test video clips - C1, C2 and C3 - that will be used for evaluating the LSPD edge detector are displayed.

Uniform Design Tests

The uniform test scheme for individual system components described in Section 3.9.2, was used for the uniform LSPD component test. The parameter settings were selected according to the uniform parameter scheme, with the aim of getting a consistent, compact and accurate combined binary result robust towards scene changes with low rendering of noise but high reproducibility of details. The parameters set found is listed in Table 4.1, while the combined binary images for all scenes tested with this set of parameters are displayed in Figure 4.7.



(a) C1 $k = 500$

(b) C2 $k = 350$

(c) C3 $k = 800$

(d) C3 $k = 850$

Figure 4.6: Test images used for evaluating the LSPD edge detector

Although the images in Figure 4.7 display some slight noise, they also show an accurate representation of detail for all scenes, with a consistent and fairly compact binary result.

Furthermore, these result show how the uniform algorithm responds to soft image objects and background regions. In Figure 4.7d, the large cluster of algae growth is represented by disperse edge points comparable to intense noise. The same observation can be made for the algae growth in Figure 4.7 (a) and (b) as well, but at a much smaller, less dominant scale. The surface of the fish in Figure 4.7a yields no particular edge response, as is also true for the background regions in all scenes. In the slightly blurred regions of Figure 4.7c, a significant reduction in detection accuracy can be seen.

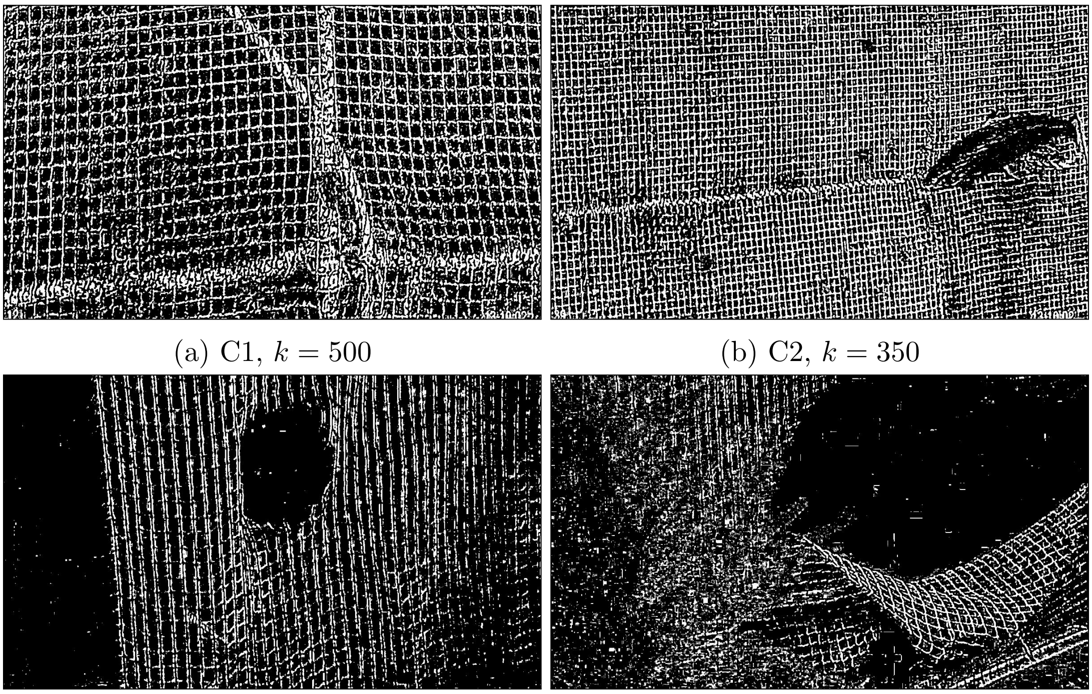
Double Direct Thresholds				
R, G, B, V	T_L	255	T_U	0
S	T_L	0	T_U	255

Gaussian Filter	
σ	3
T_s	3

Median Filter	
T_s	4

Table 4.1: LSPD optimal manually selected uniform parameters

The percentile content of edge pixels found for each individual color channel and the combined images are listed in Table 4.2. It can be observed that the percentile foreground content is fairly constant for each color channel in their respective scenes, and that the total foreground content increases as the channels are combined. This information will later be used for comparing segmentation amounts and accuracy with other algorithms.



(a) C1, $k = 500$

(b) C2, $k = 350$

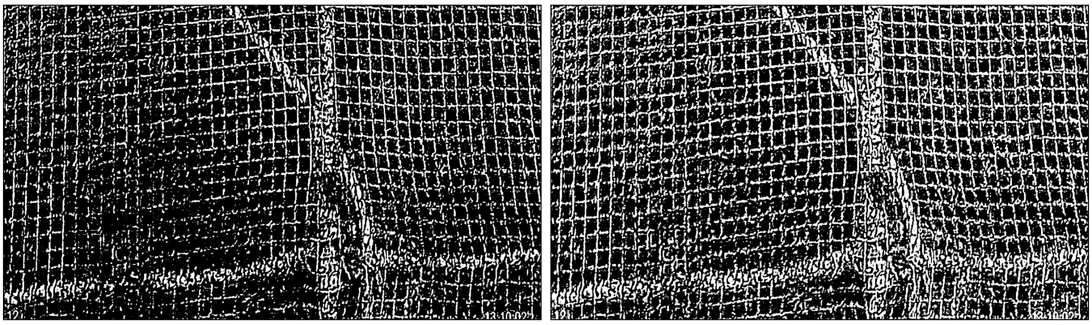
(c) C3, $k = 800$

(d) C3, $k = 850$

Figure 4.7: Combined binary images with uniform parameters

	C1 $k = 500$	**C2** $k = 350$	**C3** $k = 800$
Red	21.9 %	29.8 %	9.22 %
Green	21.6 %	30.0 %	9.20 %
Blue	20.9 %	29.5 %	9.21 %
Saturation	24.5 %	32.8 %	9.65 %
Value	21.6 %	30.0 %	9.20 %
Combined	31.6 %	39.6 %	15.17 %

Table 4.2: Percentile content of edge pixels in individual image channels



(a) C1 $k = 500$, Blue

(b) C1 $k = 500$, Combined

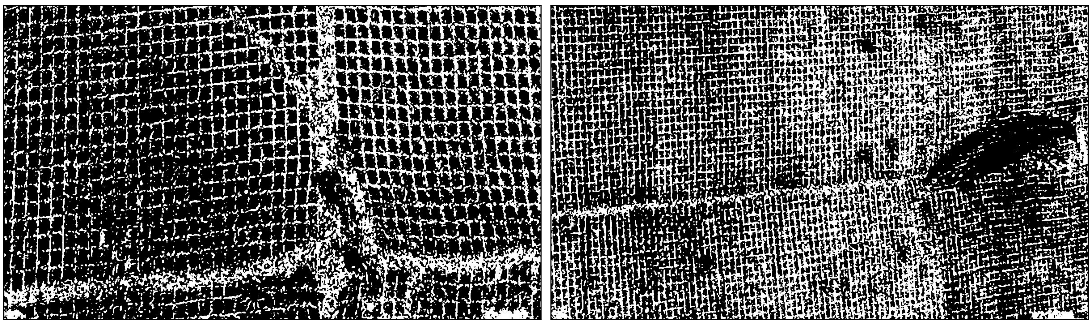
Figure 4.8: Increased detail reproduction by combining image components

An interesting observation from analyzing the uniform LSPD image channel's binary results individually, is seeing how otherwise incomplete undersegmented results combined yield an accurate segmentation. This can be seen in Figure 4.8, where the blue binary component is compared to the combined binary image. The measured percentile increase in foreground pixel content in Table 4.2 when channels are combined, coincides well with this observation.

Unique Design Tests

The unique LSPD component experiments were made according to the unique system component test scheme described in Section 3.9.2, with a unique parameter scheme. Parameters were chosen in order to give a continuous and compact binary edge image with an accurate reproduction of detail while best suppressing noise. The parameters found for the Gaussian an median filters were equal to that of the uniform experiments listed in Table 4.1, while the thresholding values for the DDT operation were considerably lower.

From all color channels analyzed with unique parameters, the saturation channel gave the better result, as illustrated in Figure 4.9. It can be seen from these images that the unique parameters for C1 featured good noise suppression and pixel compactness, but mediocre segmentation consistency. Also, the parameter setting tuned for C1 oversegmented the scene from C2, while providing poor accuracy and an inconsistent structure representation. A similar lack of parameter robustness to different scenes was found for all unique image channels.



(a) C1, $k = 500$ (b) C2, $k = 350$

Figure 4.9: Unique LSPD binary result with thresholds $T_L = 0$ and $T_U = 1$ when applied to C1 and C2 for the saturation channel

4.3.2 Comparison and Discussion

The unique LSPD parameter scheme provided no promising results in any particular aspect. The set of parameters found were highly fragile to scene changes, with a typical response being an oversegmented and inconsistent binary image. At best, the segmentation accuracy was adequate, and with low amounts of noise. If the binary results from the individual image channels were to be combined, the result would most likely be heavily oversegmented, making an LSPD module with a unique parameter scheme unsuited for the unique combinatorial design scheme. The remainder of this discussion will therefore be dedicated to the uniform LSPD component tests.

The uniform LSPD parameter set (Table 4.1) was found to work equally well for all clips evaluated, with a stable segmentation amount and accurate detail rendering. As predicted in Section 3.6, the uniform LSPD method did not respond to smooth image elements, such as fish, algae growth, blurred image regions, and background areas in any particular manner. It can be seen, however, that some structures produced a fair amount of noise, which seems to be a response to isolated sharp pixels within each structure. In general, the uniform LSPD method appeared to segment sharp foreground elements very well, with a low response to smooth elements and blurred image regions. At no point during the experimentation did the LSPD algorithm oversegment, making it seem suitable for utilization in a uniform combinatorial approach.

In light of the main goals for a modern edge detector listed in Section 2.5, one can say that the uniform LSPD algorithm displayed strong edge localization and single response characteristics, but with an adequate only optimal detection rate due to the lacking noise suppression. The binary edge results based on the images from C3 (Figure 4.7) might appear to have both misplaced edge localization and a double edge response along the net threads; however, by inspecting the original images, one will find the edges indeed should be found along the net structure, while the net threads themselves are blurred, as incurring from the extreme light conditions found in C3.

The uniform LSPD algorithm overall seems suited for complementing the robustness of the selective update scheme in the TMCSU background model, as it most likely will not oversegment, and therefore not erroneously block the pixel update of true background pixels. Furthermore, as discussed in Section 3.7.1, the incorporation of edge information will presumably increase robustness towards the pollution of foreground pixels in the background estimate - both in general and in difficult, low motion scenes - but will most likely not fully protect against stationary, smooth foreground objects.

It was questioned in Section 3.9 as to whether combining only the most distinct image information from several image channels with a uniform design scheme potentially could compete with the result of an accurately adjusted unique parameter scheme. For the LSPD component tests in this section this was found to be true, as the uniform parameter scheme provided the better segmentation results while also remaining robust to the scenes tested.

Due to the stability and consistency of the binary result from the uniform LSPD algorithm found in this section, the foreground pixel counts listed in Table 4.2 will be used a source of comparison for segmentation performance by other methods later in this thesis.

4.3.3 Conclusion and Further Usage

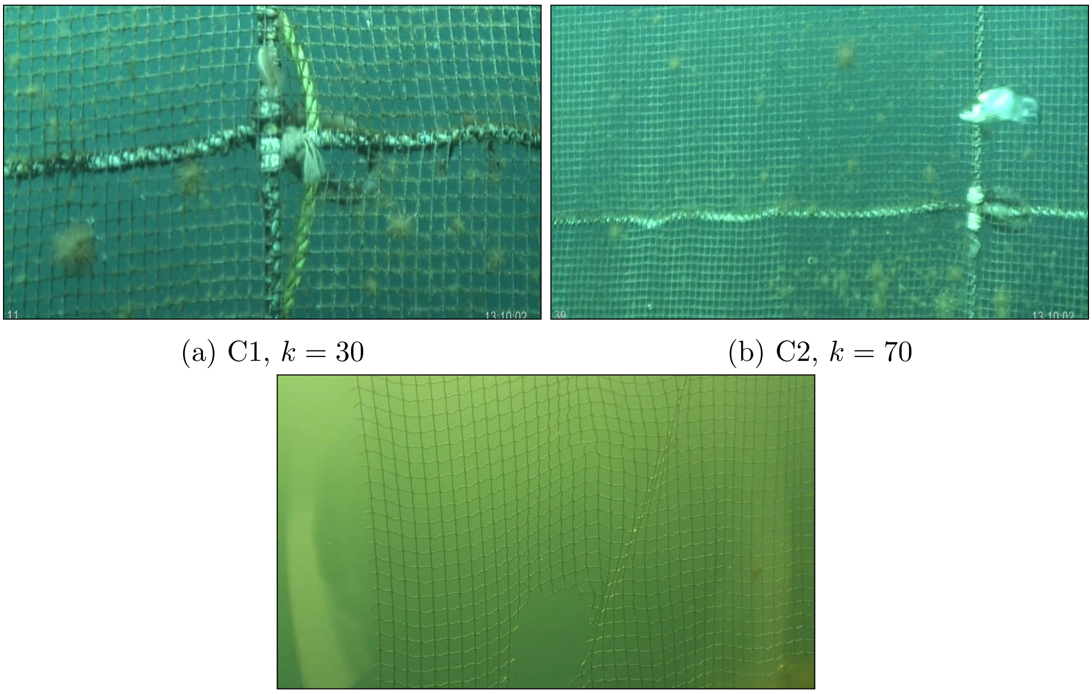
It was speculated in Section 3.9 as to whether combining multiple image channels with limited reproducibility of detail in a uniform fashion would constitute the accuracy of a single image channel with uniquely chosen parameters. In terms of the LSPD algorithm, this assumption seems to hold true. Moreover, the uniform parameter selection showed excellent robustness in all scenes evaluated - C1, C2 and C3 - with accurate segmentation results of similar characteristics for each scene, albeit with a slight presence of noise. As previously predicted, the LSPD algorithm did not detect smooth surfaces, such as fish skin or algae growth, in a consistent manner, while background regions gave no distinct response. From these results, the uniform LSPD implementation seems like a suitable component for the final system, both for improving the TMCSU model update, and as a standalone segmentation module.

4.4 Image Differencing Motion Estimation

A basic component design based on single image differencing for estimating motion was in Section 3.5 purposed. In this section, this algorithm will be tested for its ability to estimate the motion of complete net and foreground structures without over- or undersegmenting, while maintaining robustness to scene changes. The tests will be made with a unique design scheme, as well as a uniform design scheme.

4.4.1 Test Setup, Evaluation Criteria and Analysis

The aim of the experiments in this section will be to find parameter settings for the DDT binarization embedded in Figure 3.6, with the goal of achieving an accurate, consistent and robust motion estimate. Furthermore, the single differencing algorithm will be evaluated according to its applicability to the S&KB-TMSU and S&KB-TMCSU background update schemes discussed in Section 3.7.1, as well as the general possibility of using optical flow as a standalone background segmentation module in the final system design in Table 3.8. Both the unique and uniform design schemes will be evaluated, with the test setup as described in Section 3.9.2, with parameter settings either chosen uniquely for each video clip - C1, C2 and C3 - or uniformly for all clips. The test scenes that will be used for evaluating the single differencing motion estimator



(a) C1, $k = 30$

(b) C2, $k = 70$

(c) C3, $k = 275$

Figure 4.10: Test images used for evaluating the image differencing motion estimators

are displayed in Figure 4.10, all of which contain some relative motion between the camera and the background. Note that none of the binary images presented in this section have been post-processed with morphological closing.

In order to better analyze the segmentation accuracy and detection coverage of the motion estimates in this section, the original images will have their binary segmentation results overlain in a subtractive fashion; that is, if a binary point has a value of one, then this point is set to black in the original image. In this manner, the actual pixel coverage of the motion estimate can be seen, a result of very similar effect to the update blocking filter discussed in Section 3.7. When displayed in this form, the aim of the single differencing method will be to overlay every foreground point in the original image, so that only background pixels are visible in the filtered image.

Unique Design Tests

The scene specific, unique parameter selections found to best provide a consistent motion estimate for the single differencing method, are listed in Table 4.3. When applied to their respective scenes in Figure 4.10, the results found for the red image channel were as displayed in Figure 4.11. These parameter settings were selected in order to best detect the complete surface of large and small foreground objects alike, while avoiding excessive segmentation of foreground objects and erroneous segmentation of background pixels.

As predicted in Section 3.5, and distinctively visible in Figure 4.11 (a) and (e), the motion field detected is closely trailed by the previous motion estimate, with the overlapping area being excluded from the detection result. The detection coverage is quite accurate for Figure 4.11 (a) and (e), albeit with some oversegmentation in

		R	**G**	**B**	**S**	**V**
C1, $k = 30$	T_L	7	9	9	11	9
	T_U	9	5	5	11	5
C2, $k = 70$	T_L	30	30	30	25	30
	T_U	17	17	17	25	17
C3, $k = 275$	T_L	6	4	6	7	6
	T_U	6	4	6	7	4

Table 4.3: Unique DDT settings

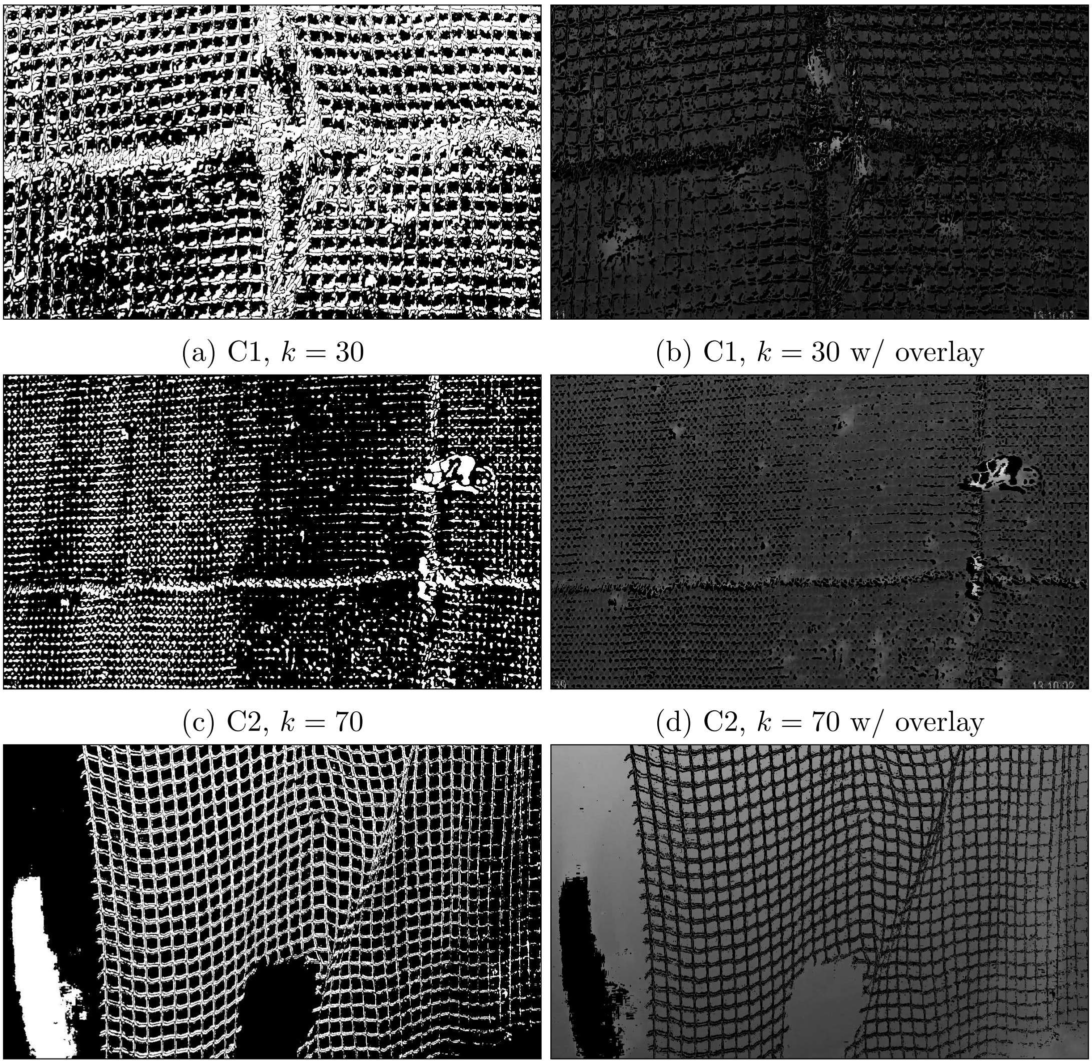
Figure 4.11a. In the medium range scene of Figure 4.11c, the foreground coverage of the motion estimate appears slightly undersegmented, while some background regions are erroneously segmented. By inspecting the uncovered regions of Figure 4.11 (b) and (d), one can see how the algae growth and loose rope ends gave a poor response in both cases. Furthermore, the algorithm's response to general, sporadic changes in brightness is visible in Figure 4.11e, where the rapidly flickering flare in the camera optics discussed in Section 1.5, has been detected as motion.

In Figure 4.11a, the red image channel does not seem to reproduce detail equally well for all foreground types, such as the darkened region and cluster of growth in the original image in Figure 4.10a. This property is further evident when comparing the red and blue binary results for this image, displayed in Figure 4.12, both of which reproduce different foreground details.

Seeing that a single image channel does not manage to reproduce some image details, the RGB & SV binary images were combined with a binary OR operation, a result of which is shown in Figure 4.13.

The combined binary images displayed the following tendencies: in Figure 4.13a, the foreground appears well covered but heavily oversegmented, except for the algae growth, which still yielded a low response; in Figure 4.13c, a slight undersegmentation is still present and some background areas are still erroneously segmented; in Figure 4.13e, the already satisfactory segmentation result gained a slightly better foreground coverage at the cost of increased noise.

The robustness of the unique parameter settings to changing scene conditions was studied by applying settings tuned for C1 to C2 and C3, the setting tuned for C2 to C1 and C3, and so on. While the parameters tuned for C1 and C2 gave similar results to their original binary images when exchanged, the exchange between C1 or C2 and C3 did not, as visible in Figure 4.14 for the combined binary images. While the parameters tuned for C1 heavily oversegmented C2, the parameters for C2 severely undersegmented C1.



(a) C1, $k = 30$

(b) C1, $k = 30$ w/ overlay

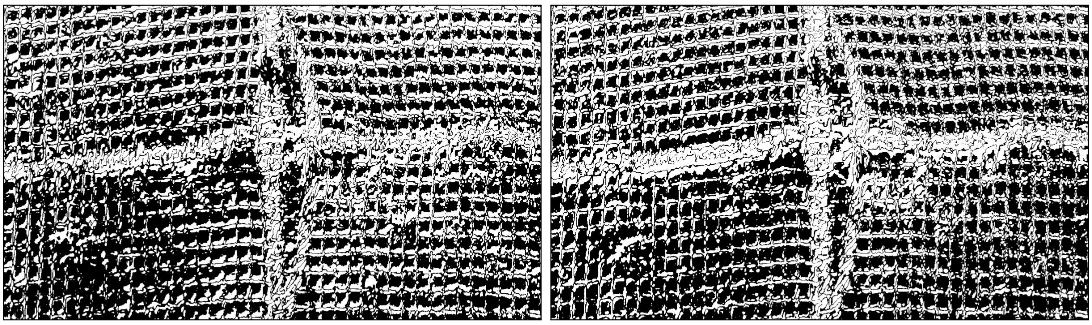
(c) C2, $k = 70$

(d) C2, $k = 70$ w/ overlay

(e) C3, $k = 275$

(f) C3, $k = 275$ w/ overlay

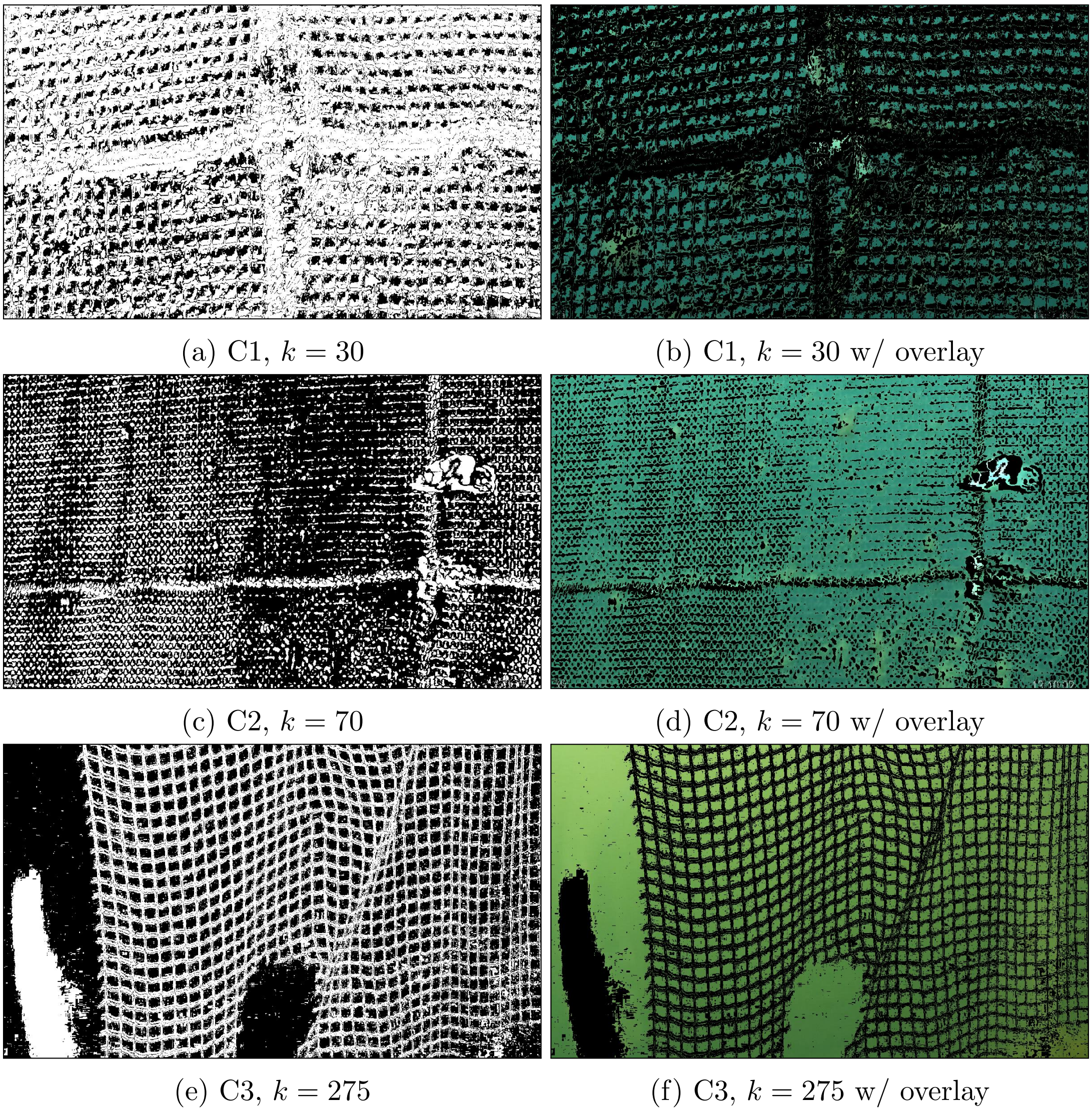
Figure 4.11: The red color channels with unique parameter settings



(a) Red

(b) Blue

Figure 4.12: Comparison of red and blue image channels for C1, $k = 30$ with unique parameters and single differencing



(a) C1, $k = 30$

(b) C1, $k = 30$ w/ overlay

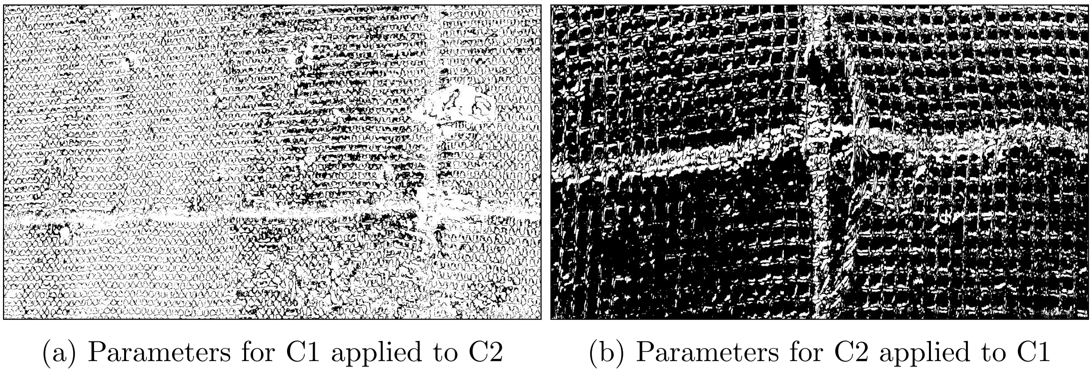
(c) C2, $k = 70$

(d) C2, $k = 70$ w/ overlay

(e) C3, $k = 275$

(f) C3, $k = 275$ w/ overlay

Figure 4.13: Binary images combined of RGB&SV image channels with unique parameters single differencing



(a) Parameters for C1 applied to C2

(b) Parameters for C2 applied to C1

Figure 4.14: Robustness of unique parameters settings to scenes not tuned for using single differencing with combined binary results

Uniform Design Tests

In order to avoid oversegmentation in the combination step associated with the uniform component design in Figure 3.16, stricter thresholding parameters were used than for the unique component tests. The C2 clip in particular, required higher threshold values than the other scenes, as using low DDT thresholds for C2 would result in heavy oversegmentation. The threshold controlling the negative histogram values for C2, T_L, would for all color channels be needed set very high, as the segmentation otherwise would inverse, displaying background as foreground, and vice versa. For these reasons, the uniform thresholding parameters were set according to the two main principles:

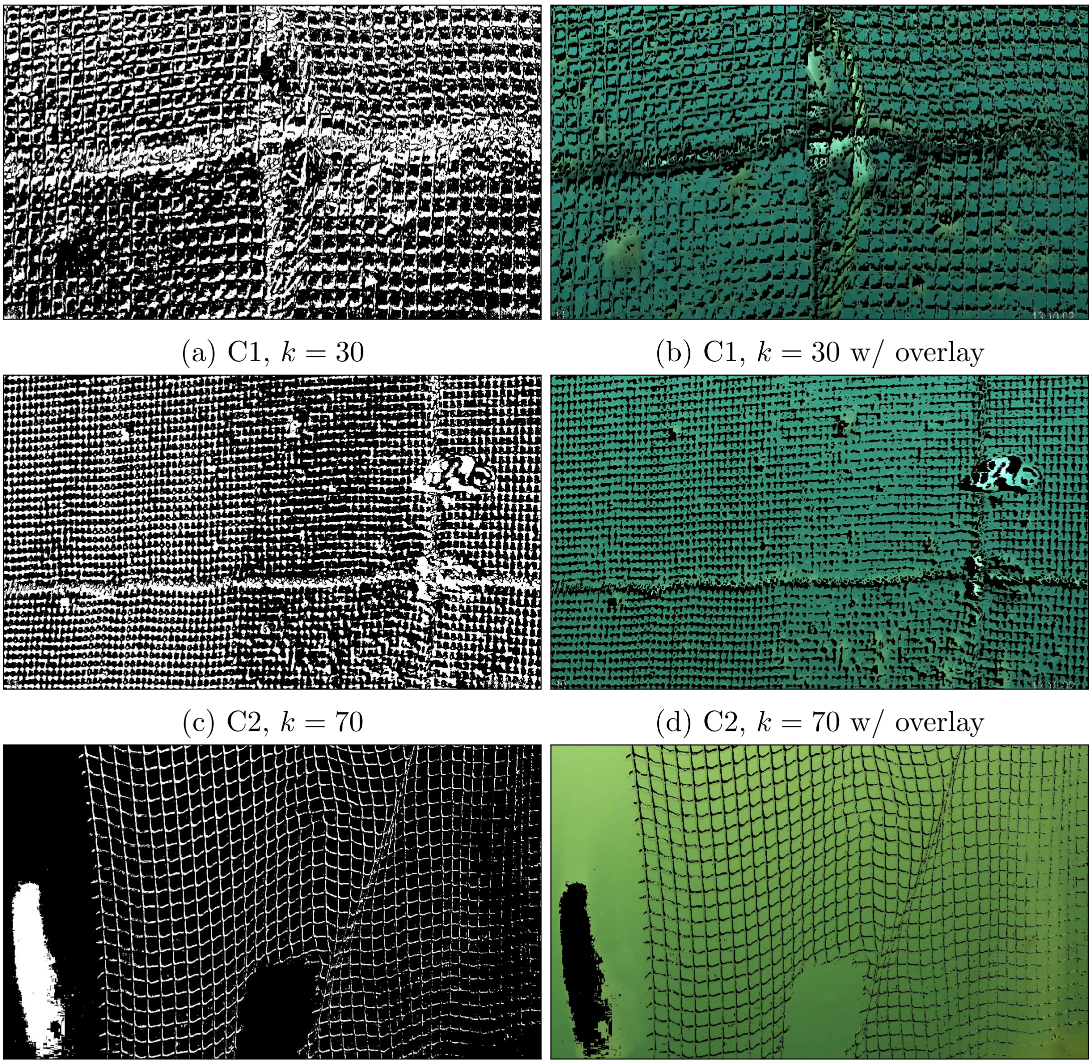
- T_L, which controls negative histogram values, was set as low as possible without inverting the segmentation result of C2
- T_U, which controls positive histogram values, was set as low as possible without oversegmenting neither C1, C2 nor C3

Based on this set of rules, the parameters listed in Table 4.4 were found. When applied to the images of Figure 4.10 there were mixed results, as can be seen in Figure 4.15.

At first glance, the combined segmentation results with uniform parameters in Figure 4.15 may seem promising. The result of C2 in Figure 4.15c in particular, which was the main deciding factor for the T_L parameter, displays a consistent segmentation with decent accuracy that covers most of the foreground object structure of the original image without erroneously segmenting background pixels. For C1 and C3, however, a lack of segmentation coverage due to the high T_L parameter is evident. In Figure 4.15e, a minimal segmentation of the net structure is presented. The segmentation is accurate, and with good consistency, but when inspecting the overlain original image in Figure 4.15f, one can see that only the structure originally brighter than the background is detected as in motion. The structure darker than the background remains uncovered by the binary overlay. The same tendency can be seen for C1 in Figure 4.15 (a) and (b). Furthermore, clusters algae growth and loose ends of the rope structure seemed to yield a low motion response for C1 and C2, as visible by the inconsistent coverage in Figure 4.15 (b) and (d); however, the growth wrapping the net threads themselves appear covered by the motion filter.

	T_L	T_U
Red	50	10
Green	45	10
Blue	50	10
Saturation	12	50
Value	55	12

Table 4.4: Uniform DDT settings



(a) C1, $k = 30$

(b) C1, $k = 30$ w/ overlay

(c) C2, $k = 70$

(d) C2, $k = 70$ w/ overlay

(e) C3, $k = 275$

(f) C3, $k = 275$ w/ overlay

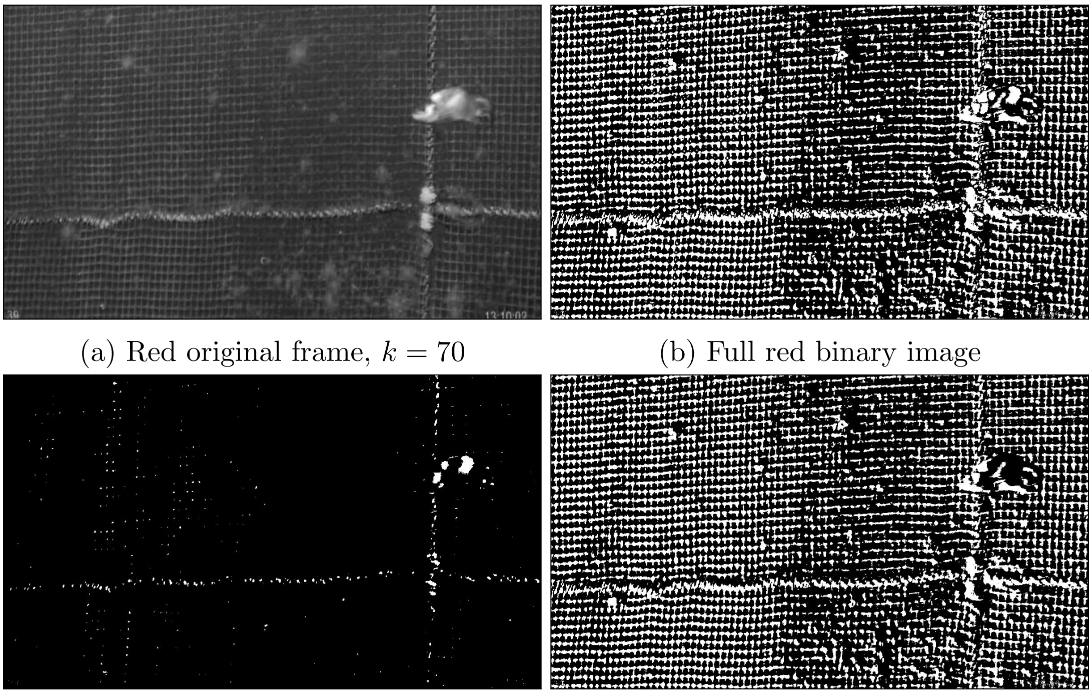
Figure 4.15: Binary images combined from RGB&SV channels with uniform parameters using single differencing

	C1, $k = 30$	**C2,** $k = 70$	**C3,** $k = 275$
Red	24.1 %	30.7 %	12.0 %
Green	26.0 %	31.4 %	10.5 %
Blue	22.3 %	28.7 %	9.03 %
Saturation	24.4 %	26.8 %	7.07 %
Value	21.6 %	27.4 %	8.12 %
Combined	36.6 %	35.9 %	13.7 %

Table 4.5: Percentile content of moving pixels in binary image with uniform parameter settings

In terms of robustness, the motion estimate with uniform parameters seemed to produce a fairly stable result for the scenes analyzed. For the motion estimates in Figure 4.15, the percentile content of moving pixels were as listed in Table 4.5. The reader might notice that these percentages are fairly similar to that of the uniform LSPD segmentation in Table 4.2.

An illustration of the extreme sensitivity to negative histogram values for the C2 scene is visible in Figure 4.16, where it can be seen that even with $T_L = 50$, the binarized negative histogram values display some erroneously segmented background pixels.



(a) Red original frame, $k = 70$

(b) Full red binary image

(c) Negative red component

(d) Positive red component

Figure 4.16: Components of the red color channel with uniform parameters

4.4.2 Comparison and Discussion

The single differencing method seem to detect spontaneous changes in the scene effectively, such as the flickering reflection from the sun seen in Figure 4.10c. Assuming that the background is slowly changing, this is a desirable feature which might be useful in terms of maintaining a robust background estimate if combined with the selective update schemes of the TMSU and TMCSU background models. A potential drawback of this feature, could be a corresponding inability to suppress image noise; however, in the video material analyzed in this thesis, this was not found to be a major issue.

Neither the unique nor uniform single differencing implementations managed to detect the motion of loose rope end and algae growth, such as those visible in Figure 4.10a. Both of these objects were found to weave with the underwater currents in a manner that frequently would make them seem stationary relative to the camera, and thereby yield a low response when analyzed with the single differencing method. The generally soft surface of these structures further complicated their detection due to a low separability from the background in each individual image channel. Based on these observations, the single differencing technique appears unsuited to detect algae growth and objects of similar characteristics.

The single differencing motion estimation algorithm with a unique parameter scheme yielded mixed results. Both the single channel (Figure 4.11e) and the combined binary results (Figure 4.13e) for the C3 scene were largely successful, with a good coverage of foreground objects, a suitable segmentation amount and a consistent performance. The single channel detection for C1 (Figure 4.11a) showed slightly inconsistent, excessive coverage, while the combined motion estimate (Figure 4.13a) was heavily oversegmented. The results from the C2 scene provided poor detection coverage of all but the most distinguishable foreground objects, while also erroneously segmenting background areas, both for the single channels (Figure 4.11c) and the combined results (Figure 4.13c). Moreover, the unique parameter settings showed low robustness to scene changes, as seen in Figure 4.14.

In general, the results from using the unique parameters scheme with single differencing motion estimation reflected several of the points discussed in Section 3.9 regarding the contradictory aims of the unique component design presented in Figure 3.14. Firstly, since every image channel should be able to sufficiently distinguish all foreground elements, parameter settings will sometimes have to compromise erroneous segmentation of background pixels with detection of dominant foreground structures, as was the case for the C2 motion estimates (Figure 4.11c and 4.13c). Secondly, when a single channel sufficiently covers most foreground objects, combining multiple channels can easily result in oversegmentation, which was illustrated by the results of C1 (Figure 4.11a and 4.13a). Conversely, the combination of image channels was also seen necessary for C1, as different channels detected motion of different objects (Figure 4.12). Finally, specifically tuning thresholding parameters for a single scenes might constrain these parameters to only work for that given scene, and an attempt to utilize these parameters for scenes not counted for will most likely result in heavy over- or undersegmentation, as was seen when exchanging the settings of C1 and C2 in Figure 4.14.

Based on these considerations, the unique parameter scheme does not seem promising for further usage with the single differencing procedure. Robustness towards scene changes is among the main goals of the system developed in this thesis, and the unique, static parameter scheme does not provide the desired level of system robustness. Later in this chapter, we will see the unique single differencing motion estimate used with the S&KB-TMSU background update, but only in the single channel, single scene fashion that this motion estimation procedure seems to manage fairly well.

One of the main motivations of the uniform parameter scheme and component design described in Section 3.9, was to increase robustness towards changing scene conditions. The uniform single differencing motion estimator seems to uphold this methodology, but at the compromise of reduced detection quality for C1 and C3, as seen in Figure 4.15. In order to avoid severe erroneous segmentation of background areas in C2 (Figure 4.15c), the threshold controlling the negative values of the difference image, T_L, had to be chosen very high, allowing only foreground elements significantly darker (more negative) than the background to be detected. This restriction crippled the detection of the net structure both for C1 (Figure 4.15a) and C3 (Figure 4.15e). The threshold controlling the positive values of the difference image could be set to levels appropriate for all clips.

The overall performance of the uniform single differencing technique was somewhat ambiguous. The motion detected for C2 (Figure 4.15c), was consistent, with a desirable coverage of the net structure, despite being at medium range. As for C1 (Figure 4.15a) and C3 (Figure 4.15e), the most distinguished segments of the net were detected, while the darker, and also smoother, sections remained undetected. And while the motion estimate enclosed these undetected thread sections in C1, both C1 and C3 showed a consistent and accurate result across their respective scenes; indeed, if the dark sections of the net threads also had been detected, the segmentation of all clips could have been considered highly successful.

If implemented with the S&KB-based update schemes of the TMSU and TMCSU background models, the regionally limited segmentation of the uniform single differencing method might prove useful in its current state. Since the foreground structures not detected by the motion estimate seems to blend fairly well with the background, they might not necessarily influence the background estimate in a destructive manner if erroneously included. Moreover, foreground objects with high contrast to the background, elements that most likely would pollute the background estimate if included, are largely detected by the uniform motion estimate. As a standalone component, however, the limited detection of this motion estimator might reduce its practical usage. On the other hand, the limitations of the uniform single differencing algorithm might be remedied if complemented with, for instance, edge information from the LSPD algorithm, which in Section 4.3 provided promising results for distinguishing net threads. Actually, the uniform system design in Figure 3.15, in combination with the S&KB-TMCSU background update scheme, utilizes this incorporation of the LSPD and uniform single differencing motion estimate, a combination which will be studied later.

None of the images presented in this section had undergone post-processing such as the morphological closing operation referred to in Section 3.5. The effect of this morphological closing will be investigated later in Section 4.7, while the results from this study will be utilized inside the S&KB-TMSU and S&KB-TMCSU modules during calculation of the update blocking filter. The single channel motion estimates from the unique single differencing method to be used during testing of the TMSU background model (Figure 4.11), will presumably benefit from this operation, as the gaps caused by the motion trail discussed in Section 3.5 most likely will be filled. In order to fill the gaps in the uniform motion estimates (Figure 4.15), however, a larger closing template might be needed, which could result in oversegmentation of the binary image. Attempting to fully close this gap might therefore not be beneficial.

Just like the LSPD method, the uniform single differencing procedure produced a fairly stable segmentation amount, as listed in Table 4.5. This data will be used for discussion later.

4.4.3 Conclusion and Further Usage

The applicability of a basic technique like single differencing for estimation of optical flow was questioned in Section 3.5. One particular remark was made as to whether the characteristic motion trail of this procedure would imply a beneficial or destructive effect to the motion detected when applied to objects of various sizes with different motion properties. Both a unique and uniform component design was investigated for this estimator, neither of which showed any particular complications with this feature; quite on the contrary, this motion trail seemed to increase the detection area, giving the motion estimate a better coverage of the foreground structures in motion. In general, the unique component design displayed a lack of robustness to scene changes and an overall inability to produce both single channel and combined motion estimates of usable quality. For these reasons, the single differencing motion estimator with a unique component design was deemed unsuited for practical usage. The uniform component design showed promising robustness to scene changes, and a consistent and accurate segmentation, but with a limited detection capability of net threads in two of the scenes analyzed. A combination of measures where the LSPD would be used to complement the limitations of the uniform single differencing motion estimator was suggested, a design which is resembled by the S&KB-TMCSU model update module, and uniform component design introduced in Section 3.7.1 and Section 3.9, respectively. Neither the uniform nor unique implementations of the single differencing technique managed to sufficiently detect the motion of algae growth and objects of similar character, a segmentation task that will have to be managed by some other system component.

4.5 Temporal Median with Blind Update

The TMBU background model was introduced in Section 3.7.1 as a measure for provide material for isolated component tests later on, and to inspect the effects of parameter settings later to be investigated with the TMSU and TMCSU background models. In particular, the effects of the sampling interval, $\triangle t$, and buffer size, n, in terms of stability, smoothness and computational complexity, will be documented. Furthermore, an approximation of the "ideal" background estimate will be made.

4.5.1 Test Setup, Evaluation Criteria and Analysis

A general set of aims was in Table 3.4 listed for the temporal background estimation methods discussed in Section 3.7.1. For the TMBU model, which employs a blind update scheme, point three does not apply directly. Instead, the background estimate will be evaluated according to its smoothness and pureness: the visual amount of non-background pixels.

The parameter combinations investigated with the TMBU background model are listed in Table 4.6, where the sampling frequency $m = \triangle t \cdot T_f$ has been used instead of the sampling interval to account for the varying frame rate, T_f, of the video clips - C1, C2 and C3 - analyzed (Table 3.3). If the length of the video clips (Table 3.2) are compared to the required initialization period of the TMBU algorithm (Equation 3.5) with the parameter combinations listed in Table 4.6, one may notice that none of the video clips contain enough frames to fully initialize the expanding buffer of the TMBU model for some of the parameter combinations listed; for instance, the combination $m = 10$ and $n = 100$ requires 1000 successive frames in order to initialize, which is nearly twice the size of all clips analyzed. In order to fully initialize the buffers for all parameter combinations, the video clips were looped during simulation.

	$n = 10$	$n = 50$	$n = 100$	$n = 200$
$m = 1$	x	-	-	-
$m = 10$	x	x	x	x
$m = 50$	x	-	-	-

Table 4.6: Parameter combinations tested with the TMBU algorithm

In Figure B.4, B.5 and B.6, the background estimates for the parameter combinations tested have been documented for C1, C2 and C3, respectively. All of the background estimates illustrated were captured at the same video frame during the last round of looping. The runtime performance of these simulations has been listed in Table 4.7.

n	m	**Total Time**	**FPS**	**Rounds**
10	5	1031 sec	1.50	3
10	50	342 sec	4.52	3
10	10	610 sec	2.53	3
50	10	843 sec	1.83	3
100	10	1003 sec	1.54	3
200	10	>7200 sec	N/A	6

Table 4.7: Runtime performance of the TMBU background model for C1

The parameter combination with $n = 200$ did not complete simulation. After about 1050 frame iterations, at which point the expanding buffer had a size of approximately $n = 105$ frames, the simulation progress staggered, and at the two hour mark the whole simulation was manually canceled. It was registered that the process consumed roughly 7.0 GB of memory at the time of cancellation, which combined with other processes depleted the available memory of the computational platform.

Altering the Sampling Frequency

With a fixed size buffer, changing the sampling frequency, m, appeared to augment of every characteristic naturally inhibited by the background model in terms of learning rate, initialization period, robustness, as well as its responsiveness to scene changes, indirectly also affecting the appearance of the background estimate.

For C1 and C2 (Figure B.4 and Figure B.5), changing the sampling frequency displayed no effect on the smoothness or pureness of the background estimate. Instead, by increasing the sampling frequency (by lowering m), the degree of which the background estimate rendered a continuous trail of the previous positions of distinct objects in the scene increased. By lowering the sampling frequency (increasing m), image points contained by this trail of previous motion was instead scattered across the background estimate.

A similar tendency was visible for C3. For the approximately 10 frames prior to the scene illustrated in Figure B.6, a scene similar to that of Figure B.3i was briefly displayed in C3 (some of the scenes in C3 were drastically shortened by the error correction conducted in Section 3.2.2). As a result of these abrupt scene transitions, the algae growth of the previous scene still persisted in the background estimate of the settings with a higher sampling frequency (Figure B.6b), while the lower sampling frequency settings displayed no response the abrupt changes (Figure B.6d and B.6f).

In terms of the computational complexity, it can be seen from Table 4.7 that increasing the sampling frequency increased the processing required, and vice versa.

Altering the Buffer Size

With a fixed sampling frequency, altering the buffer size, n, seemed to greatly influence the smoothness and pureness of the background estimate, and also the required memory of the computational platform.

As can be seen for all clips in Figure B.4, B.5 and B.6, increasing the buffer size from 10 to 20 radically improved the smoothness and pureness of the background estimate, an effect that increased until about $n = 50$, where these qualities stabilized. The adaptability, or learning rate, of the background model equally decreased for a larger buffer size, and for $n > 50$, the background did not change, even for C3, which featured swift scene changes. Also, the computational required increased, as seen in Table 4.7, but the most noticeable increase was the RAM requirements of the buffer structure.

4.5.2 Discussion and Further Usage

The size n of the temporal median buffer appears to directly control the model's initialization period and adaptability to changing surroundings, and conversely its robustness towards foreground objects erroneously included in the model estimate, as well as the pureness and smoothness the background estimate. While a small buffer may easily visualize foreground elements erroneously included in the background estimate, the same objects might be neglected by the median operator for a larger buffer. Moreover, a larger buffer would take longer to adapt to new scenes and initialize, and would additionally have a larger computational demand in terms of memory and processing power. Furthermore, the larger buffer size, seemed to improve the smoothness and pureness of the background estimate.

From the observations made in the previous section, one might say that the sampling interval, $\triangle t$, or its sampling frequency counterpart, m, essentially controls the rate of which data is fed into the background model. By reducing the sampling interval, data flow per time unit into the model will increase - and vice versa. While any buffer setting n will require a pre-determined amount of frame data to initialize and adapt, increasing the data flow effectively increases the learning rate and decreases the initialization period for the given setting. Moreover, the system responsiveness to quick scene changes appears to increase with a higher data flow, at the cost of additional computational demands; since there will be less change in the scene between each sampled frame for shorter sampling intervals, the impact each scene element has on the background estimate as a whole is equally increased: for instance, stationary foreground objects that erroneously influence the background estimate will have a greater negative effect with a shorter sampling interval, while correctly updated image regions likewise will have a greater positive effect. In summary, the sampling interval $\triangle t$ augments most positive and negative characteristics of the system, and might desirably be chosen as short as possible within the limitations of the computation platform for better responsiveness, while balancing system robustness and learning rate controlled by the buffer size.

From Table 4.7, it can be seen that the required processing power for a sampling frequency of $m = 5$ is close to that of a buffer size of $n = 100$. Since evaluating only every 50th frame would lead to a highly unresponsive detection system, as for the $m = 50$ setting, it seems like adjusting the sampling frequency and buffer size generally has a low influence on the required processing power for operating the temporal median buffer structure. The real computational limitations, however, seemed to be the extreme memory requirements that was found for the larger buffers. In a practical implementation, a buffer size of $n = 100$, requiring as much as 7.0 GB of computer memory, does not seem like a viable option, in particular if an untethered ROV is to be developed. Following these considerations, further tests with the TMSU and TMCSU models will focus on buffers of less computational complexity, that is, in terms of memory requirements.

When comparing the original frames to their background estimates in Figure B.4, B.5 and B.6, one will see that the estimates approximates the overall light conditions of the original scenes very well, such as the gradient transitions for C1 and C2, and the bright halos in C3. As such, the TMBU model, and the TMSU and TMCSU models which carry the same buffer structure and statistical operator as the TMBU model, will probably tolerate soft light conditions equally well on a general basis, a desired feature to ensure detection robustness in difficult light conditions.

While the pureness and smoothness of the estimates increases for larger buffer sizes, none of the background estimates are truly pure. This might be caused by the large presence of foreground pixels relative to background pixels in the videos analyzed. During the analysis of the LSPD and single differencing methods in Section 4.3 and 4.4, it was estimated that roughly 35% of the image consisted of foreground pixels for C1 and C2, while C3 contained about 14% foreground pixels. When comparing these percentages to the amount of pureness for the $n = 100$ background estimates in Figure B.4, B.5 and B.6, it seems to fit that a lower amount of foreground increases the pureness. Intuitively, this would also indicate a possible issue with calculating an accurate background estimate if the amount of foreground raises too high, or more precisely above 50%, as the model then might estimate the foreground as background, and vice versa. In practical terms, this relation could indicate that a inspecting a double net structure might be of difficult due to the correspondingly high amount of foreground structure, and also that cleaning the net before inspection could improve the background estimate, since less foreground structure would be present. Whether these concerns will be an issue with the S&KB-TMSU and S&KB-TMCSU update schemes, however, is another matter, since these update schemes actively will attempt to filter foreground objects from the background estimate.

The background estimates with $n = 100$ will be utilized for further component testing as an approximation of the ideal background estimate for the C1, C2 and C3 scenes, due to their overall smooth, homogeneous and pure characteristics. However, with the introduction of the update blocking filter in the TMSU and TMCSU background models, the erroneous inclusion of foreground objects might not be as evident as for the TMBU estimates, even with smaller buffer sizes and lower sampling intervals. Such background estimates will therefore also be analyzed, in particular for investigating the effect of a background smoothing filter in Section 4.6, with the aim of reducing the roughness associated with lowering the buffer size.

Furthermore, the evaluation of the parameters found with the TMBU model will be used to narrow the search for suitable parameters for the TMSU model. The tests will focus on smaller buffer settings and low sampling intervals, due to the overall beneficial system characteristics emerging from such a selection. If the introduction of the selective S&KB-TMSU background update scheme performs as intended, the low pureness of the background estimates found with these settings when using the TMBU model might not be a concern.

4.5.3 Conclusion

In this section, the main characteristics of the TMBU model, the effects of its parameters, and the limitations the background estimate calculated has been investigated. The sampling interval and buffer size parameters of the TMBU model both seemed to offer a compromise between a desirable learning rate, a short initialization period, and a high responsiveness on one side, versus robustness towards dominant foreground structures, smoothness and pureness of the background estimate on the other. Moreover, the beneficial factors from the first contradicted that of the other. The computational processing requirements were fairly constant for either parameter combination, but the memory requirements elevated markedly for larger buffer structures. Ultimately, a parameter range believed to fit TMSU model for further testing was predicted, and general guidelines as to how the parameters should be set found. Furthermore, several considerations of the usability of temporal background segmentation in practical terms was discussed, and image material for further component tests collected.

4.6 Background Subtraction and Smoothing

Based on the background estimates calculated with the TMBU background model in Section 4.5, a set of component tests and parameter evaluations for later usage will in this section be conducted. Both the TMSU and TMCSU background models rely on the calculation of a binary foreground image, either through background subtraction and DDT binarization directly, as for the TMSU model in Figure 3.11, or firstly smoothing the background estimate, followed by subtraction and DDT binarization, as for the TMCSU model in Figure 3.12. Since the background subtraction step is the last operation of all the temporal background segmentation methods discussed in Section 3.7.1, an evaluation of general challenges and properties of this operation will be discussed, points that largely motivated the design of the TMSU and TMCSU models, as well as the uniform combinatorial design in Figure 3.15.

Due to the assumed increased roughness of the background estimates produced by the TMSU and TMCSU models from using a buffer of limited size, smoothing of the background estimate before subtraction was suggested in Section 3.7.1. For this reason, both the ideal[1] and rougher background estimates from the TMBU experiments will be analyzed. By comparing the segmentation performance of a rough and an ideal background estimate, the importance of maintaining a smooth and accurate estimate can be analyzed, as well as investigate the further usability of the TMSU and TMCSU methods.

While the TMSU model will not be evaluated according to its robustness properties towards different scenes, the TMCSU model will. For this reason, both a unique and a uniform parameter scheme will be analyzed.

[1] the smoothest and least polluted background estimate found from the TMBU experiments

4.6.1 Test Setup, Evaluation Criteria and Analysis

The aim of all the tests conducted will be to find some measure of binarizing the difference images from the background subtraction step of the temporal background segmentation methods in Section 3.7.1 in order to provide a robust, accurate and properly segmented binary foreground image. For the DDT technique used, both a unique and uniform component design will be tested. However, only the uniform design will employ a smoothing operation prior to the binarization. This division derives from the order of which the experiments were finished, as illustrated by the time steps in Figure 4.1, but also the further usage of the methods: while early experiments with the TMSU model used a unique parameter scheme, the TMCSU model focused on robustness towards different scenes, a task not suited for static unique parameter scheme, as will be seen shortly. The frames that will be used as input of the background subtraction (Equation 3.4) can be seen in Figure 4.17, while the background estimates, calculated using to the buffer settings in Table 4.8 with the TMBU method, can be found in Figure 4.18.

No outside system components rely on the performance of each individual image channel for any of the temporal background segmentation algorithms, as seen in Figure 3.15 and 3.14 . Therefore, the accuracy of the segmentation for each individual image channel will not be investigated for neither the unique nor uniform parameter schemes. It is, however, still of interest that no single channel misbehaves, ultimately leading to the degeneration the combined segmentation result.

	n	m
Ideal	100	10
Rough	10	5

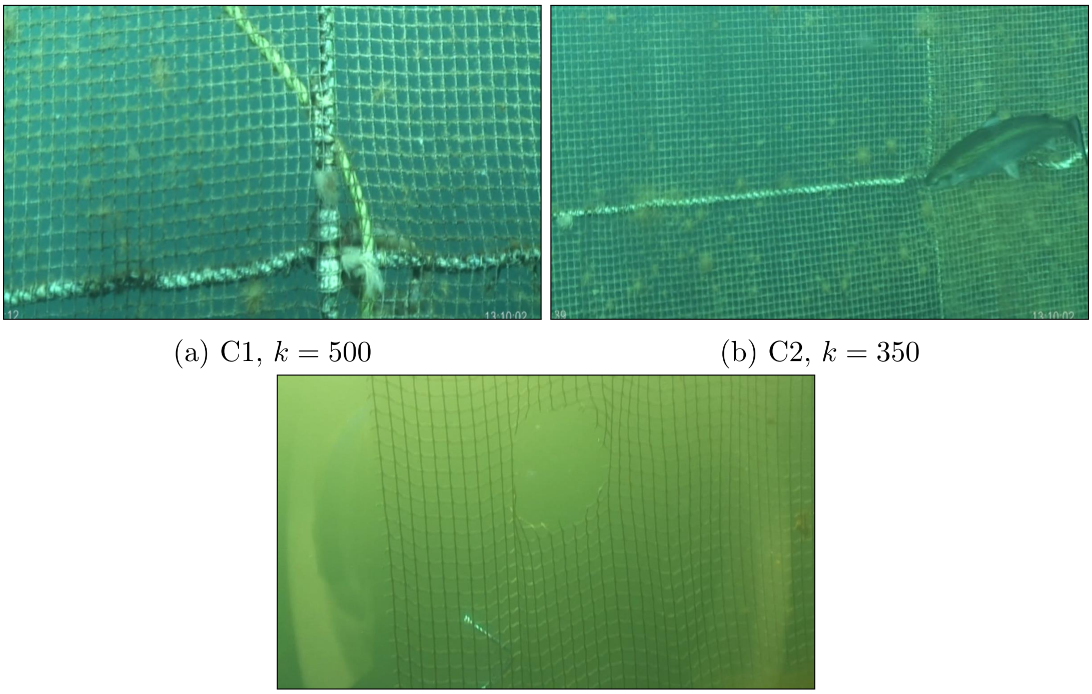
Table 4.8: Ideal and rough TMBU background estimate settings

Unique Parameters

Both the ideal and rough background estimates in Figure 4.18 were analyzed for finding a suitable unique parameter set. The unique DDT thresholds were selected as low as possible without oversegmenting or inverting the segmentation, with the final selection as listed in Table 4.9. Binary foreground images corresponding to this set of parameters can be found in Figure 4.19. No smoothing or post-processing was made to the images of the unique parameter tests.

One can see from Figure 4.19 that algae growth, and most other foreground elements, is solidly segmented for C1 and C2 for both the rough and ideal background estimates. Fish, however, appears to only yield a partial response, as seen in Figure 4.19c and 4.19d.

The segmentation result from using the ideal background estimate with C3 displayed low reproducibility of net the threads, as seen in Figure 4.19e. Also, this segmentation shows the effect of the flickering bright halo present in C3, as described in Section 1.4. Since this the halo is impermanent, it will here be considered a foreground element, and not a segmentation error.



(a) C1, $k = 500$

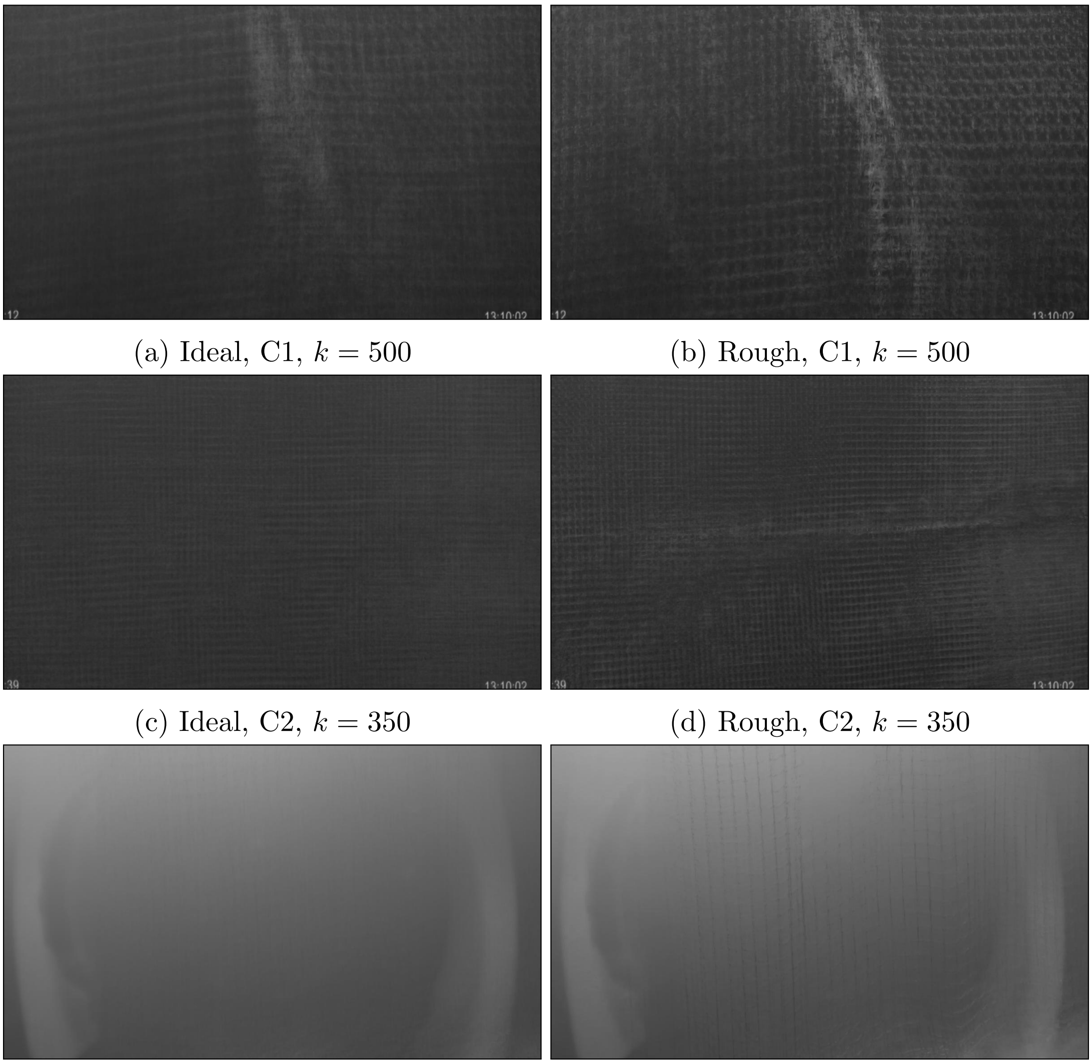
(b) C2, $k = 350$

(c) C3, $k = 450$

Figure 4.17: Test images used for background subtraction

		R	**G**	**B**	**S**	**V**
C1, $k = 500$	T_L	40	30	7	0	25
	T_U	5	10	15	50	10
C2, $k = 350$	T_L	30	30	30	0	30
	T_U	0	0	0	40	0
C3, $k = 450$	T_L	6	5	5	5	6
	T_U	7	10	5	8	10

Table 4.9: Unique DDT settings for background subtraction



(a) Ideal, C1, $k = 500$

(b) Rough, C1, $k = 500$

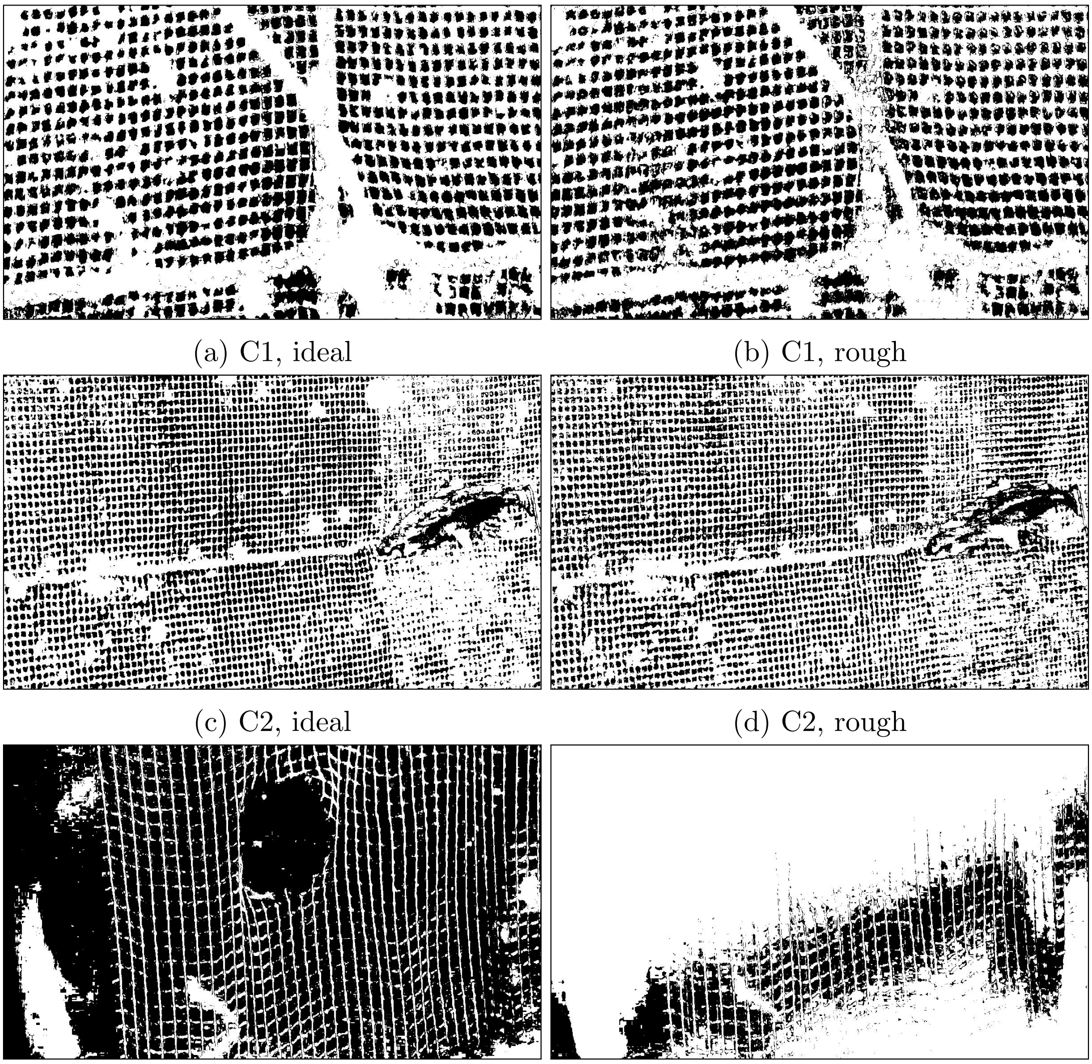
(c) Ideal, C2, $k = 350$

(d) Rough, C2, $k = 350$

(e) Ideal, C3, $k = 450$

(f) Rough, C3, $k = 450$

Figure 4.18: Rough and ideal background estimates of the red channel from the TMBU method. Rough setting: $n = 10$, $m = 5$. Ideal settings: $n = 100$, $m = 10$.



(a) C1, ideal

(b) C1, rough

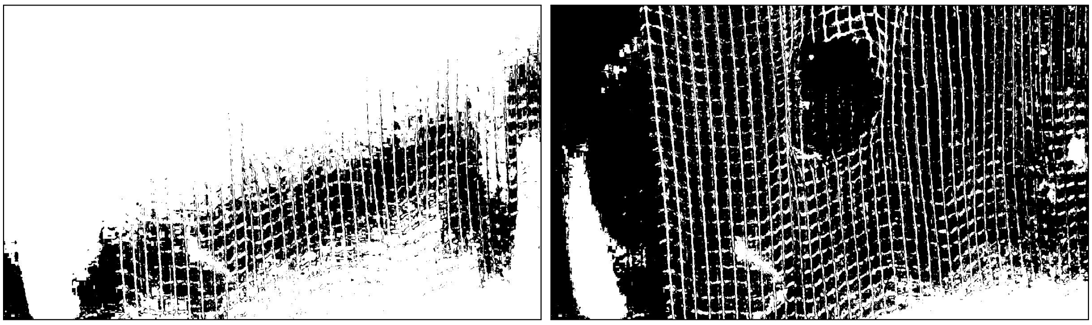
(c) C2, ideal

(d) C2, rough

(e) C3, ideal

(f) C3, rough

Figure 4.19: Combined images with unique parameter settings on ideal and rough background estimates



(a) Combined w/ value channel

(b) Combined w/o value channel

Figure 4.20: C3, $k = 450$ combined segmentation result with and without value channel on rough background estimates using unique parameter settings

The result from the rough C3 background estimate in Figure 4.19f is vastly over-segmented. However, it also illustrates the potential degenerative effect on a combined binary image when an individual image channel misbehaves. By inspecting the combined binary image from this background estimate with and without the value channel, that is, the combined binary image from RGB&SV and RGB&S respectively, a misbehavior of the value channel becomes clear, as seen in Figure 4.20.

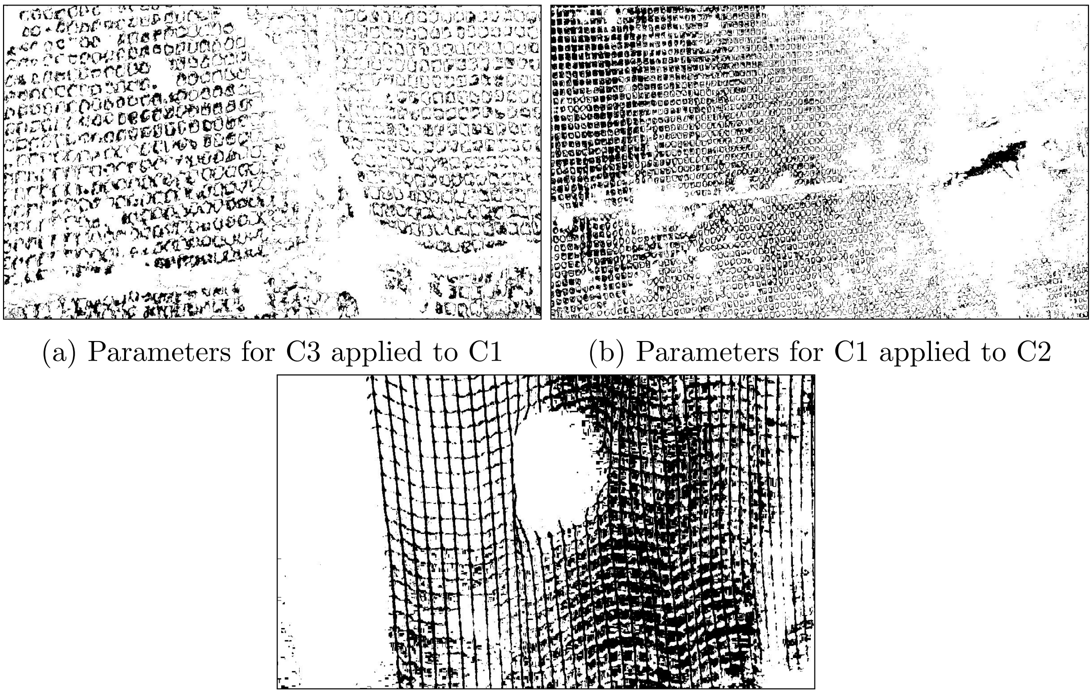
Apart from the misbehavior of the value channel in Figure 4.19f, which most likely was a caused by an inaccurate background estimate, as will be discussed soon, there seems to be little difference between the segmentation results based on the ideal and rough background estimates. Most notably, some of the particularly coarse areas in the rough background estimates in Figure 4.18 resulted in the segmentation of background pixels.

Robustness of the unique parameter set in Table 4.9 towards scene changes was investigated by exchanging the thresholds particularly tuned for one clip to the video of the other clips. As has been illustrated in Figure 4.21, none of the unique parameter settings functioned for any other scene than the one they were specifically tuned for.

Like the both for the LSPD and single differencing modules, the binary segmentation result was different for each image channel in the background subtraction. In Figure 4.22, it can be seen how the red channel better distinguishes algae growth, while the blue channel better reproduces shaded net threads and ropes.

Uniform Parameters and Smoothing

The effect from applying a smoothing operation on the background estimate before subtraction has been visualized in Figure 4.23, where the rough, red background estimate from C1 (Figure 4.19b) has been subtracted and binarized with and without the pre-smoothing of a median filter. The template size used was $T_s = 30$, which was selected based on the filters ability to eliminate disperse erroneous segmentation. To better see the results of the smoothing, the upper DDT threshold was set to $T_U = 255$ to ignore all positive histogram values, while the lower threshold was set to $T_L = 20$, a setting that for the red channel provided an inverse segmentation, as seen in Figure 4.23b. In the segmentation result from the smoothened background estimate, one can see the effect of the median filer, where dispersed, smaller regions of the inversed segmentation have been suppressed, while the compact region in the middle has become



(a) Parameters for C3 applied to C1

(b) Parameters for C1 applied to C2

(c) Parameters for C2 applied to C3

Figure 4.21: Combined images with exchanged unique parameter settings

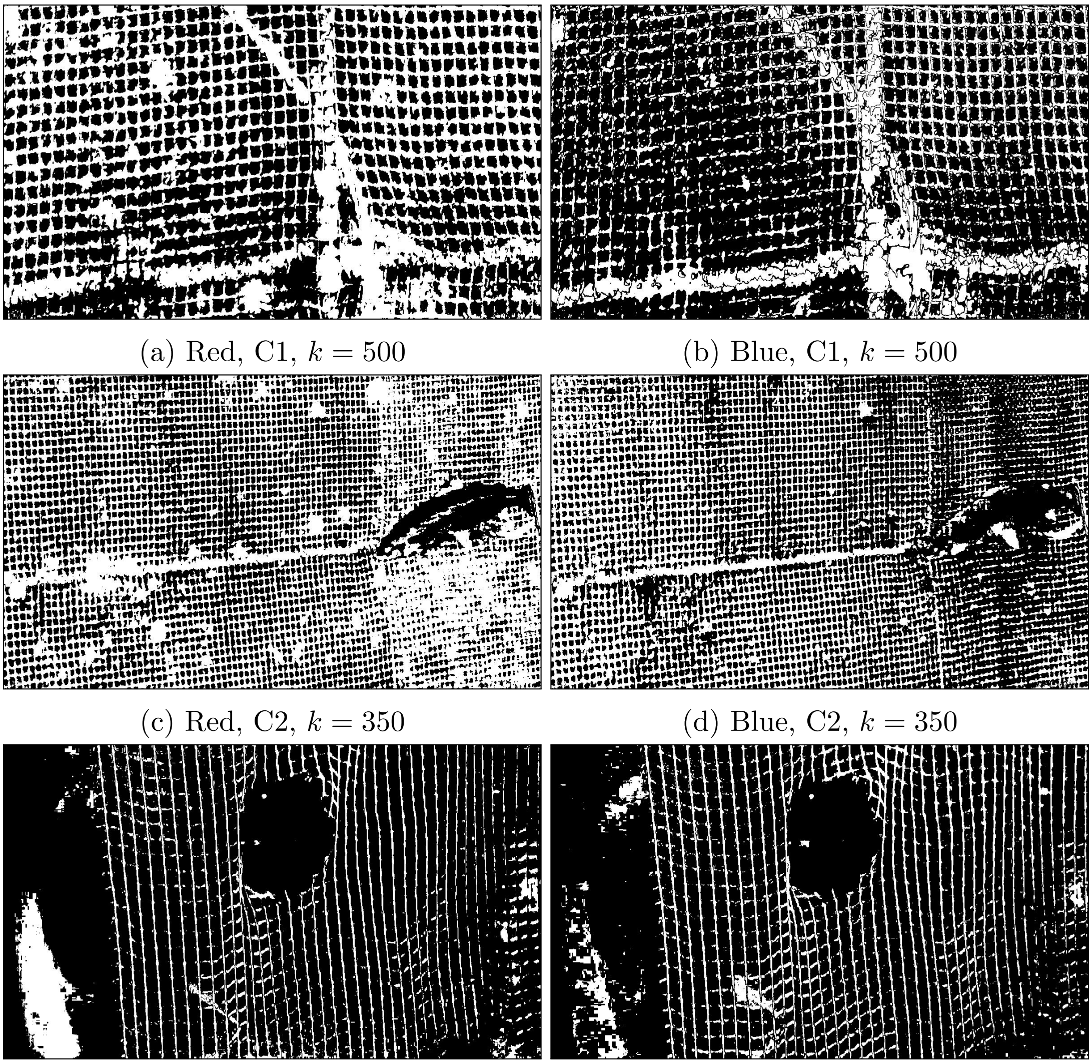
more consistent.

The uniform DDT parameters were selected based on the median filtered background estimates from Figure 4.18. When inspecting the parameters for the unique DDT binarization in Table 4.9, a large gap in the value ranges used for the scenes in C1 and C2 and the scene in C3 was noticed; in fact, several thresholds for C2 were as much as six times higher than that of C3. Furthermore, while the lower thresholds were higher than the upper thresholds for C1 and C2, the reverse was true for C3. A suitable parameter combination for C1 or C2 would therefore most likely inverse the segmentation of C3, as was seen in Figure 4.21c. A significant compromise was therefore required when the uniform parameters were selected. For most of the uniform thresholding parameters listed in Table 4.10, the highest of the thresholds for each clip in Table 4.9 was used as a starting reference, while the smoothing effect of the median filter allowed for a reduced threshold for some image channels without reversing the segmentations or oversegmenting the binary image components. The final combined binary results from using the smoothing filter setting and uniform DDT parameters listed in Table 4.10, are illustrated in Figure 4.24.

When utilizing a uniform parameter set on the smoothened background estimates from Figure 4.18, very similar tendencies to that of the unique combined results in the previous section were found: for C1 and C2, most structure, and in particular algae growth, was clearly and accurately segmented, while the fish was not. Moreover, the value channel in Figure 4.24f misbehaved also for the uniform parameter set, but with a slightly less severe extent.

The compromise made in the parameter selection was clearly visible in Figure 4.24e, where the higher threshold values suppressed most foreground elements for C3.

When comparing the segmentation results calculated of the smoothened ideal and



(a) Red, C1, $k = 500$

(b) Blue, C1, $k = 500$

(c) Red, C2, $k = 350$

(d) Blue, C2, $k = 350$

(e) Red, C3, $k = 450$

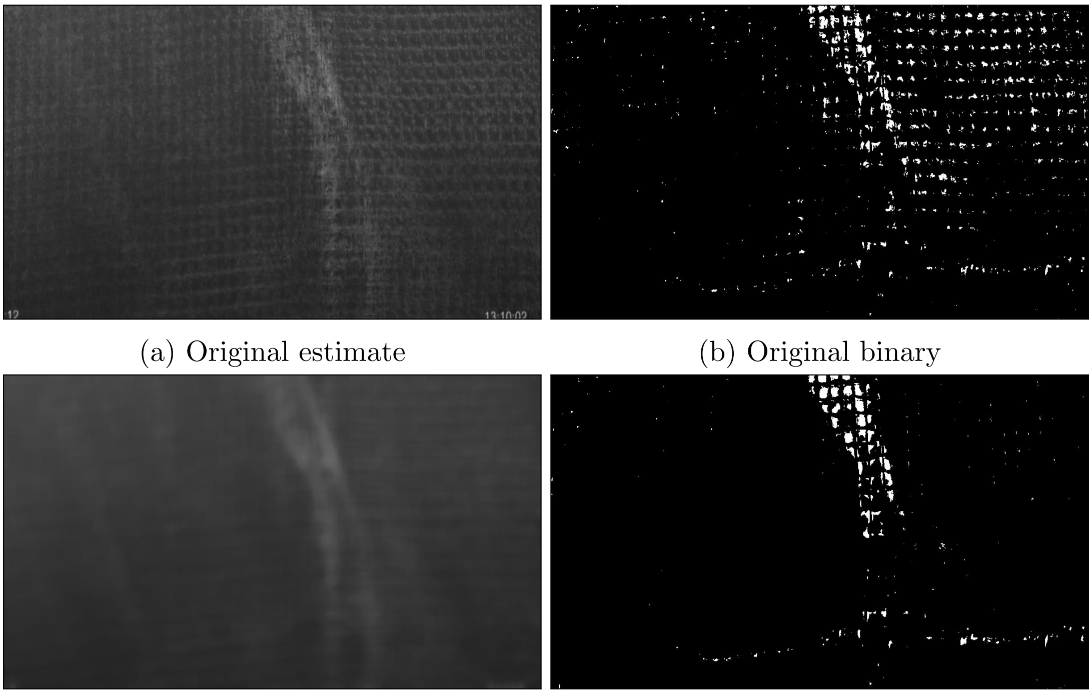
(f) Blue, C3, $k = 450$

Figure 4.22: Difference in background segmentation results with the red and blue channels using unique parameters on ideal background estimates

	T_L	T_U
Red	30	7
Green	30	10
Blue	30	18
Saturation	5	35
Value	30	10

Median Filter	
T_s	30

Table 4.10: Uniform DDT and background smoothing parameters



(a) Original estimate (b) Original binary

(c) Smoothened estimate (d) Smoothened binary

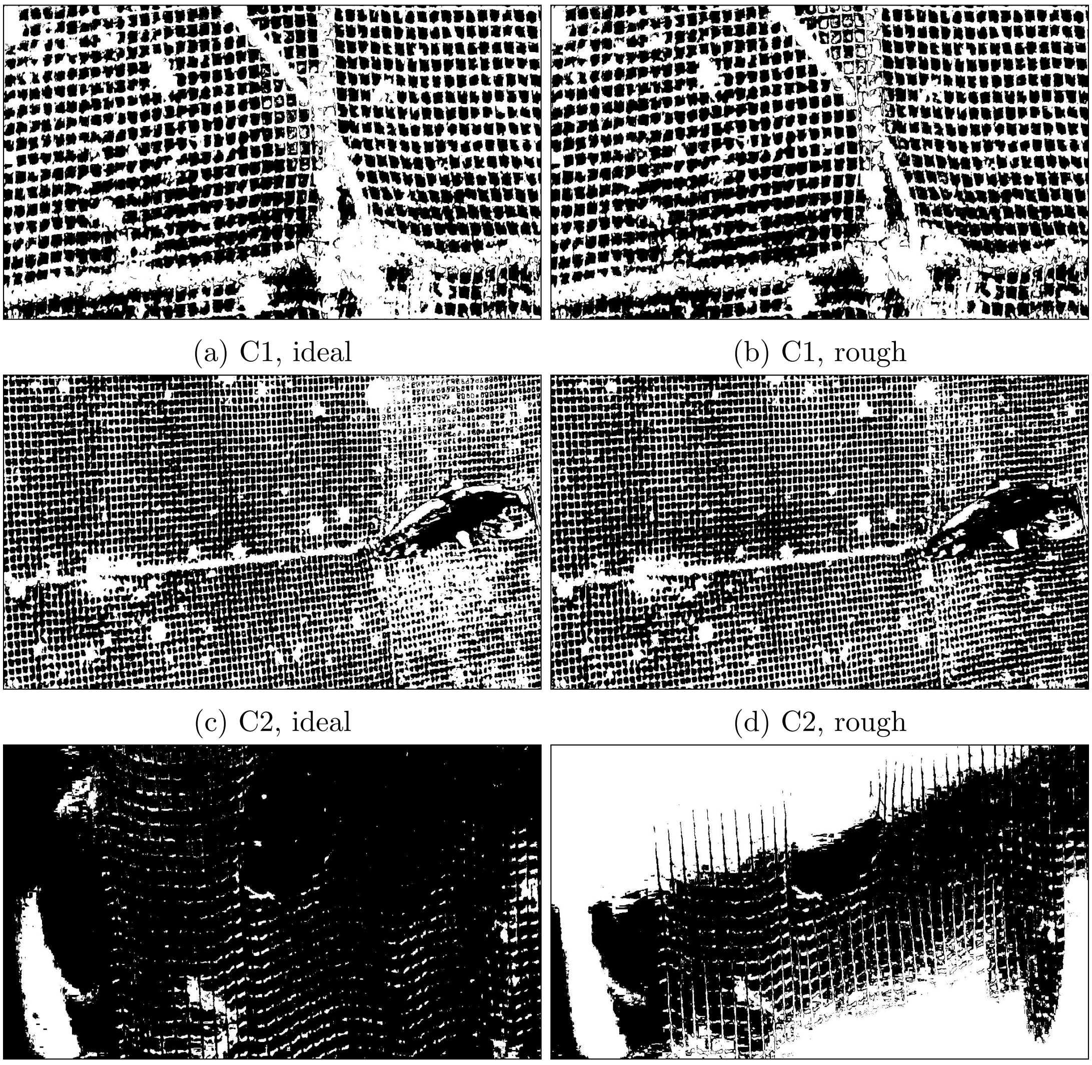
Figure 4.23: Effect of smoothing a rough background estimate before subtraction using a median filter with $T_s = 30$, and DDT $T_L = 20$ and $T_U = 255$

rough estimates for the C1 and C2 in Figure 4.24, it appeared to be only minor differences in separating the two. While the fish and net structure in the ideal C2 segmentation appears slightly better reproduced than the rough segmentation result, the ideal and rough binary images for C1 were of seemingly identical characteristics.

4.6.2 Discussion: Further Usage and Considerations of Temporal Background Segmentation

From a general analysis of the quality of the combined binary segmentation results found with unique and uniform parameters illustrated in Figure 4.19 and 4.24 respectively, a few observations can be made. The unique parameter scheme seemed to overall provide a more accurate and consistent segmentation result than the uniform setup, in particular for C3 in Figure 4.19f, where the scene specific unique parameters yielded a much better reproduction of the net structure. Both parameter schemes appeared to segment net structure, ropes and algae growth in a desirable manner, while neither managed to completely segment moving fish in a reliable manner. Compared to the single differencing and LSPD methods previously investigated, neither of which segmented algae growth in a distinctive manner, the consistent segmentation of algae growth is a welcome feature of the segmentation result from the background subtraction method.

In terms of robustness, the background subtraction with unique parameters showed heavily oversegmented and inversed results, as seen in Figure 4.21. For this reason, the unique setup seems unsuited for anything by experimental, single scene analysis, which it indeed will be used for during in the TMSU experiments in Section 4.8.



(a) C1, ideal

(b) C1, rough

(c) C2, ideal

(d) C2, rough

(e) C3, ideal

(f) C3, rough

Figure 4.24: Combined images with uniform parameter settings and smoothing on ideal and rough background estimates

The uniform parameter scheme, on the other hand, showed robustness towards scene changes, but at the compromise of a greatly degraded detection accuracy for C3, as seen in Figure 4.24e, due to incompatibility of threshold values between the scenes C1 and C2 versus C3. With a limited applicability, a temporal background segmentation method implementing the uniform component design in Figure 3.15 would therefore have to rely on the robustness of other standalone segmentation modules to identify the remaining foreground elements in scenes where the uniform parameters fail.

Although not all occurrences of fish in the analyzed video streams yielded a poor segmentation response with the uniform and unique subtraction schemes, the inferior segmentations displayed for C2 in Figure 4.19 and 4.24 indicate that temporal background segmentation alone cannot be used to reliably segment fish in the final system. In Section 4.4, the single differencing motion estimate was found to quite accurately segment fish on multiple occasions, and might therefore function well in redundancy with temporal background segmentation for segmenting fish. Since the appearance of fish can vary greatly depending on how the ambient lightning reflect from its skin, using both temporal and motion based background segmentation methods combined might manage to segment most, if not all, fish when combined.

The temporal background segmentation appeared to not handle all scenes equally well with uniform parameters, as can be seen from Figure 4.19e and 4.24e where the net structure was largely rejected due to too high thresholding values. It is presumingly still a viable approach for the analysis of such scenes based on its ability to detect algae growth general foreground objects, but not by itself. The LSPD method, however, displayed robust and accurate segmentation results for net threads in all scene analyzed in Section 4.3, but inferior ability to detect algae growth. Combining the segmentation results from the temporal background segmentation and LSPD methods might therefore benefit the final binary result.

The ideal and rough background estimates provided segmentation results of similar quality. Without the smoothing operation of the median filter, the ideal estimate provided a cleaner segmentation, that is, without the moderate erroneous segmentation found from utilizing the rough estimate more visibly polluted from foreground pixels, as seen in Figure 4.19. With the smoothened background estimates, however, the difference was hardly visible, as displayed in Figure 4.24, even though some of the threshold values were lower for C1 and C2 than for the original background estimates in Figure 4.19. It appears that employing a median smoothing filter on the background estimates not only allowed for lower threshold values without degrading the segmentation result, as seen in Figure 4.23, but also largely closed the distance between ideal and rough segmentation results in terms of segmentation quality, as can be seen in Figure 4.24. Furthermore, the median operation seemed to cluster isolated areas of erroneously segmented background pixels due to estimate impureness into localized blocks, overall reducing the potential area of effect of the erroneous segmentation. In other words, when utilizing a uniform parameter scheme with median background smoothing, the smoothness of the background estimates appears to have a low impact on the resulting binary segmentation, while the degrading effect of pollution in the estimate is centralized.

It was mentioned Section 3.7 that an outdated or poorly maintained background estimate might potentially oversegment large image regions. This was exemplified by the misbehavior in Figure 4.20, where the rough background estimate for the value channel most likely was not fully adapted to the current scene at the time, possibly due to some recent scene transition characteristic for C3, as discussed in Section 4.5.

Since the roughness of the background estimate appears to be of less importance when employing a uniform parameter scheme with background smoothing, the main remaining task of the TMSU and TMCSU models, as well as the uniform combinatorial system design, would be to: handle the impurity in the background estimate while retaining a real-time computable background model structure with low memory requirements; the incorporation of multiple segmentation methods to better manage the shortcoming of each individual segmentation method; and also to control the misbehavior of individual image channels in the temporal segmentation methods, as was seen for both the unique an uniform parameter schemes in this section, as illustrated in Figure 4.20.

The main purpose of the TMSU model will be to reduce the pollution of foreground pixels in the background estimate, while experimenting with a rougher estimate of low memory requirements, as a forerunner to the TMCSU model. In a final implementation, the closed single image channel calculation of the blocking filter in the S&KB-TMSU update scheme, illustrated in Figure 3.11, would most likely suffer from the limited segmentation ability found for all single image channel segmentation methods investigated in this and earlier sections. While a unique parameter scheme potentially could boost the single channel performance, the TMSU as the main temporal background segmentation method would still lack robustness towards scene changes. Mainly for these reasons, the TMSU model will only function as a step towards implementing the TMCSU model, and not as a component of the final system.

The uniform component design in Figure 3.15, which incorporates the TMCSU model, attempts to address the remaining concerns discussed in this section. By combining the segmentation results from the LSPD and single differencing techniques in a combinatorial uniform scheme, where both methods function as standalone segmentation components while also supporting the calculation of the update blocking filter through the TMCSU method, the individual limitations of each segmentation method and the limited rendering capabilities of individual image channels will potentially be alleviated. Furthermore, since a uniform parameter scheme is employed with this combinatorial module design, robustness to scene changes is encouraged. Moreover, in the uniform system design, the segmentation results from the individual TMCSU channel calculations will be combined with the overshoot exclusion technique described in Section 3.9.3, which will attempt to exclude image channels that severely deteriorate the combined binary results, as was experienced both for the unique and uniform parameter schemes with the value channel in Figure 4.19f and 4.24f.

4.6.3 Conclusion

In this section, background subtraction with a unique and uniform design, as well as the possibility of background smoothing, has been studied. The unique parameter scheme performed better in terms of segmentation quality, but showed highly unfavorable results in terms of robustness. While displaying robust behavior for different scenes, the uniform parameter scheme greatly compromised the segmentation quality for net threads in certain scenarios. Both rough and ideal approximations of the background scene were utilized as background estimates, to which the effect of employing a median smoothing operation before background subtraction was investigated. Based on the resulting quality from these experiments, image roughness seemed to be a low influencing factor in the final segmentation result, while pollution from foreground pixels showing presence in the background estimate still remains a source of error. Furthermore, the potential degenerative effect of a single image channel misbehaving due to an outdated background estimate was documented. Temporal background segmentation was in general found excellent at distinguishing algae growth in the analyzed video material, but with a situational performance for fish and net threads.

Temporal background segmentation may largely be considered the cardinal tool of this thesis, and the background subtraction step covered in this section is the final step in all the temporal background segmentation methods analyzed. For this reason, several concerns of background segmentation in general has been discussed, concerns which has motivated the development of the TMSU and TMCSU models, the uniform combinatorial component designs, as well as the single differencing motion and LSPD edge detectors investigated in this thesis.

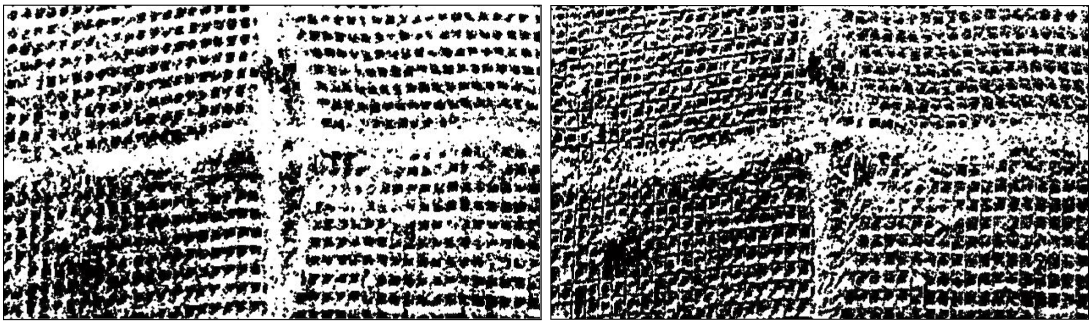
4.7 Morphological Operation Parameters

Morphological closing operations are embedded into the S&KB-TMSU and S&KB-TMCSU model update schemes presented in Figure 3.13, where binary foreground, motion and edge images are post-processed before the update blocking filter is calculated. The intention of this operation is to improve the consistency of the foreground structures in the binary segmentation images, as discussed on several occasions through this report. In this section, a brief summary of the closing parameters found to work best for each respective binary image and closing operation conducted will be given.

For all binary images studied, the aim was be to find a functioning closing operation that improved the consistency of the binary foreground structures without closing entire net masks or oversegmenting the binary image. The binary images for C2 were found to be the limiting factor in most cases, as larger structuring elements easily would close net masks entirely for this scene.

Since the net structure is viewed from multiple angles and at different orientations, the shape of the structuring element was chosen to be a disc; while a square might seem like a natural choice considering the squared meshes of the net, the disc was chosen due to its orientation independent behavior.

In Table 4.11, an overview of the radii used for the disc structuring element in the various closing operations in the S&KB-TMSU and S&KB-TMCSU update schemes illustrated in Figure 3.13 is given.



(a) Figure 4.11a closed with $R = 2\,[px]$ (b) Figure 4.15a closed with $R = 3\,[px]$

Figure 4.25: Single differencing binary images after closing operation: **a**. Unique parameters red channel **b**. Uniform parameters combined image

	S&KB-TMSU	**S&KB-TMCSU**
Foreground	2 $[px]$	0 $[px]$
Single Differencing	2 $[px]$	3 $[px]$
LSPD	-	1 $[px]$
Combined image	3 $[px]$	3 $[px]$

Table 4.11: Radii of the disc-shaped morphological closing structuring elements used in the S&KB-based update schemes

It was questioned in Section 4.4.2, as to whether a closing operation would fill the segmentation results of the unique and uniform single differencing method. As can be seen in Figure 4.25 for their respective radii, R, smaller gaps were filled for both methods, while larger gaps remained open.

4.8 Temporal Median with Selective Update

The TMSU background model was introduced in Section 3.7.1 as an approach for excluding moving objects classified as foreground from polluting the background estimate during model update, a feature absent in the TMBU model. Potential benefits of this selective update was to allow for a less computationally expensive buffer structure of smaller size that, when compared to a larger buffer structure of the TMBU model, managed to maintain a background estimate of equal quality while providing fast initialization and a high learning rate.

Since the TMSU background model can be considered an iteration of the TMBU background model with a selective update consolidated by the single differencing motion estimate, the main properties of the TMSU has already been covered in Section 4.4 and 4.5 during investigation of the single differencing method and the TMBU model, respectively. For this reason, only the additional properties of the TMSU model will be studied, which are: properly utilizing the background preservation rate, w_b, characteristic for the S&KB update scheme, and investigate the benefits and challenges of the S&KB update scheme in terms of pollution of the background estimate, learning rate, initialization period. For better analysis, the frame indicators from Section 3.8 will be utilized. Most findings from this section will carry directly to the TMCSU model to be employed in the final system in Section 3.11.

4.8.1 Test Setup, Evaluation Criteria and Analysis

The aim of the parameter selection was to find a combination that balanced initialization period, learning rate and responsiveness versus robustness towards semi-stationary foreground objects well. In order to find a functional combination of the buffer size, n, sampling frequency, m, and background preservation rate, w_b for the S&KB-TMSU update scheme, two tests were conducted. Firstly, an array of parameter combinations based on the observations made for the TMBU model in Section 4.5 - selected in a hit-or-miss fashion - were visually inspected, from where a seemingly lucrative combination range was found. Secondly, the resulting limited parameter range was closely logged and compared for each clip tested, until a suitable combination for all clips was discovered. The initial test parameter combinations can be found in Table B.1, the limited range is summarized in Table 4.12, while the combination that was found to function for all clips while preserving a promising balance of the features desired is given in Table 4.13.

	$n \rightarrow$	**10**	**15**
$w_b \downarrow$	(m)		
2		(5,10)	(5,10)
5		(5,10)	(5,10)

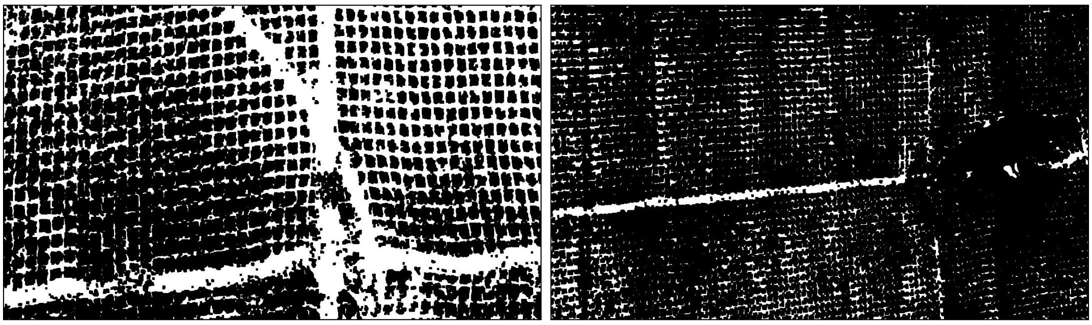
Table 4.12: Limited parameter setting test range for the TMSU model

n	m	w_b
10	5	2

Table 4.13: Universal parameter setting for the TMSU method

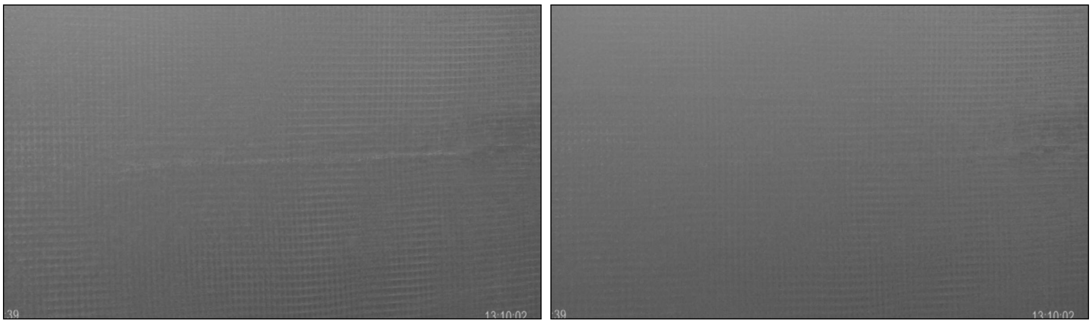
In order to analyze the continuous performance of the simulations made with TMSU model, the frame quality indicators introduced in Section 3.8 where utilized for graphing purposes. For all indicators, a running mean of the previous $Q = 50$ values was used, as this setting seemed to display most tendencies well. Graphs containing all frame indicators for the TMSU simulations can be found in Appendix B.3.2.

In Figure B.7, background estimates for the clips C1, C2 and C3, which was calculated using the parameters in Table 4.13, can be seen. By comparing these estimates to those calculated by the TMBU method at equal points in the clips with equal buffer settings (see Figure B.4, B.5 and, B.6 for $n = 10$ and $m = 5$), it can be seen that while the roughness still remains in TMSU background estimates, the impurity is of different character. For C1 and C3 in Figure B.7, the foreground pollution seems lessened compared to their respective TMBU images; however, the pollution also appears



(a) C1, $k = 500$ (b) C2, $k = 350$

Figure 4.26: Comparison of net coverage of the blue channel update blocking filters for C1 and C2

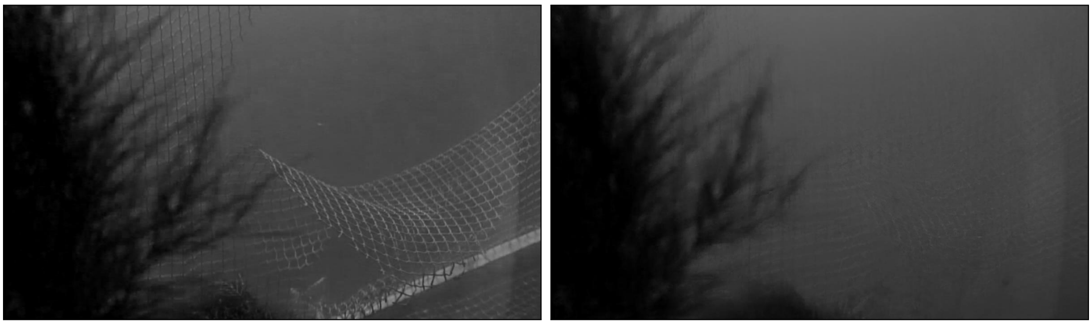


(a) $w_b = 2$ (b) $w_b = 5$

Figure 4.27: Increasing the background preservation rate for $n = 10$ and $m = 5$, as seen for C2, $k = 350$, blue channel

slightly denser, as the "motion trail" found for the TMBU equivalents no longer is present. The difference is less visible for C2, which appears to be resulting from a limited update blocking filter segmentation; when comparing the update blocking filters of C1 and C2, as displayed in Figure 4.26, it can be seen how the filter for C1 covers most the net threads, while the filter for C2 does not.

Increasing the background preservation rate generally seemed to improve the smoothness and robustness of the background estimate in most cases, as seen in Figure 4.27; however, an increased preservation rate equally seems to lower the learning rate and prolong the initialization time of the algorithm, as can be seen by inspecting the graphs in Figure B.11 and B.12. While the count of undefined pixels (the black pixel count) in the graph with $w_b = 2$ drops at $k = 21$, the setting with a slightly higher $w_b = 5$ drops at $k = 36$, practically meaning that the $w_b = 2$ setting provides a functioning background estimate 15 frames, or 58%, faster than the $w_b = 5$ setting. They both settle at close 0% of undefined pixels at approximately $k = 75$, at which point they can be considered fully initialized. Furthermore, one can see how the lower w_b adapts quicker to scene changes by inspecting the faster change in amplitude of the percentile content of foreground pixels in Figure B.11 and B.12. The parameter setting of Table 4.13 was closely contended by the equal setting with $w_b = 5$ in most aspects, making it a logical source of comparison, and an alternative setting for the TMSU method with a higher focus on background preservation.



(a) C2, $k = 500$ (b) C2, $k = 500$ estimate

Figure 4.28: Dominant, stationary algae growth influencing the blue TMSU background estimate

By comparing the graphs in Appendix B.3 to the behavior of the segmentation result and motion estimate during visual inspection of the TMSU simulation, it seemed like the binary foreground image provided results of favorable segmentation quality when the percentile content of foreground pixels followed the movement of percentile pixels in motion. Similarly, the binary foreground image for each respective image channel was found to heavily oversegment on multiple occasions when the percentile content of one image channels was significantly higher than that of the others, as can be seen when comparing Figure B.13 and B.14 for the red and blue channels, respectively; at frame $k = 325$, for instance, the blue segmentation result contained 82% foreground pixels, nearly twice that of the red segmentation result. Furthermore, the amount of sharp, or edge, pixels seemed to correspond well with the amount of foreground elements in the scene.

A particular context in which the TMSU model consistently seemed to develop an erroneous background estimate, was in scenes where dominant clusters of algae growth were slowly moving or stationary, as seen in Figure 4.28.

4.8.2 Discussion and Further Usage

Although not explicitly explained or illustrated in this section, results from both video analysis and multiple parameter settings will here be discussed.

It appears that increasing the background preservation rate, w_b, greatly prolonged the initialization time and reduced the learning rate, but equally increased robustness to stationary objects. Conversely, decreasing w_b seemed to greatly shorten initialization time and increase the learning rate, but decrease robustness to stationary objects. The effect of adjusting the w_b parameter was in general also found greater than modifying the buffer size n in terms of these properties. Consequently, the added memory requirements from increasing w_b for a desired system characteristic appeared more lucrative than increasing the buffer size, since both increased the number of frames to be calculated, but the background preservation rate provided a stronger effect per included frame. Still the importance of a properly sized buffer should not be underestimated, as the quality of the repeated background frames entirely depended on a good current background estimate, which originally was determined by the buffer size n.

The single buffer parameter setting in Table 4.13 was deemed a functional combination suited for all clips analyzed. However, this does not imply that no other parameter combinations were better for different situations; in fact, all the combinations tested for in Table 4.12 yielded desirable results, balancing initialization time, learning rate, robustness and system responsiveness in various manners. The setting in Table 4.13 merely combined the right amount of each property found to work better for the vastly different scenes and challenges of all clips combined, such as the swift scene transitioning in C3. Therefore, a practical implementation of the S&KB-based updated scheme, which is also found in the TMCSU model, would most likely benefit more from some other similar parameter setting to that given in Table 4.13. In general, the background preservation rate was found to be a fairly intuitive parameter with a conclusive effect on the actual system performance, well suited for fine tuning an eventual implementation.

The results from the TMBU and TMSU are very similar, in particular for C2; however, this might be caused by the relatively poor performance of the unique single differencing method in distinguishing net threads, as was found in Section 4.4. In scenarios where the update blocking filter managed to cover all foreground elements well, the background estimate was of equally increased quality, as can be seen by comparing Figure B.6b and B.7f from the TMBU and TMSU background estimates, respectively. For the TMCSU model, which employs a more advanced blocking filter, this is a promising result.

By inspecting the performance graphs found with the frame indicators in Section 3.8, as displayed in Appendix B.3, the overall behavior of the TMSU simulation could be analyzed in detail. Although the exact "optimal" percentages were different for each clip analyzed, the fluctuation and relation between the indicators allowed for offline determination of segmentation quality, occurrence of channel misbehavior, initialization period and learning rate, as well as a stable relative reference of the total amount of foreground pixels in the scene. For analysis of the final system, and further usage, these indicators show great potential. The graphs form the TMSU simulations specifically will be used to set the limit for the overshoot exclusion used in the uniform design scheme in Figure 3.15.

The relatively low memory usage associated with the small sized buffer structures investigated in this section makes the S&KB-TMSU, and also the S&KB-TMCSU model, seem applicable to a decentralized computational platform, such as an AUV, in future implementations.

4.8.3 Conclusion

In this section, the forerunner of the TMCSU model, the TMSU model, has been investigated. Through a series of experiments, the characteristics and potential benefit of the S&KB selective update scheme has been analyzed, and a set of parameters for usage with the TMCSU model found. It was seen that exclusion of foreground elements during update of the background estimate lessened pollution, but that the actual performance heavily depended on the quality of the update blocking filter. Furthermore, the convenient analytical capabilities of the frame indicators from Section 3.8 were displayed.

4.9 Combining Binary Images: Overshoot Exclusion

The overshoot exclusion method was in Section 3.9.3 introduced as a simple illustrative measure for managing the occasional oversegmentation that can occur when a temporal background segmentation method utilized a outdated or not fully adapted background estimate, a scenario where combining binary images with an OR operation might be unsuitable. Since the TMCSU, which utilized the overshoot exclusion, follows from the TMSU model, the overshoot limit, L, will be chosen from the performance graphs of the TMSU model simulation in Appendix B.3.

Since the average percentage of foreground pixels differs for C1, C2 and C3, L was set slightly higher than the highest foreground percentage found during normal operation for all clips. The limit selected can be found in Table 4.14.

L	70 [%]

Table 4.14: Overshoot Exclusion limit found for the TMSU and TMCSU segmentation modules

Chapter 5

Experimental Results: Final System

In this chapter, the experimental results from the final system design of this thesis, as summarized in Section 3.11, will be evaluated. The evaluation will consist of two steps: firstly, the main uniform combined background segmentation module, named "Background Segmentation Process" in Table 3.8, will be analyzed; and secondly, the binary foreground images from the first will be inspected using the "Damage Assessment" setup from Table 3.8.

The performance of each individual subcomponent of the final system was analyzed and discussed in Chapter 4. For this reason, the individual performance of internal system components in the final system will not be evaluated in this chapter. Instead, the combined performance of the modules, and their total ability to ultimately provide a comprehensive and robust background segmentation result for the damage detection algorithm will be studied. The aim of the damage detection algorithm, and thereby the final system, will be to generate a binary damage image suitable for a future implementation.

Unlike for the experiments in Chapter 4, the discussion and concluding thoughts on the performance of the final system will be covered in separate chapters, that is, in Chapter 6 and 7, respectively.

5.1 Background Segmentation Process

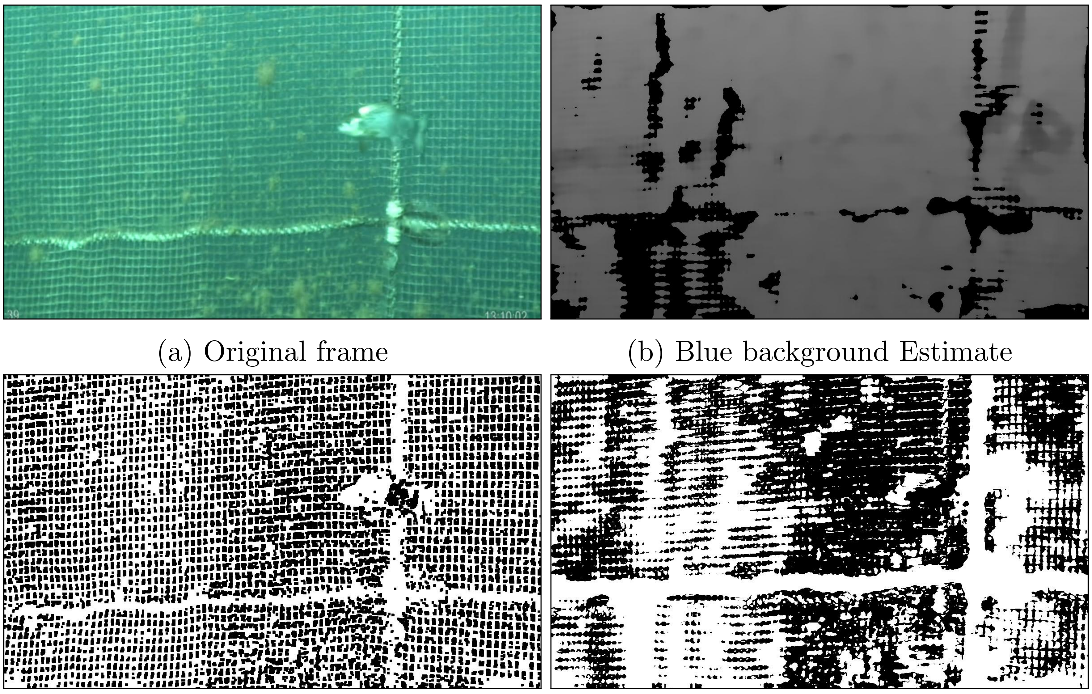
The background segmentation process of the final system is represented by the uniform combinatorial design scheme in Figure 3.15. Among the (uniform) segmentation modules incorporated, only the added complexity of the update blocking filter in the TMCSU background model has not already been fully investigated in Chapter 4. Results regarding the S&KB-TMCSU update scheme, as well as the general properties and behavior of the combined uniform system will therefore here be given. All results displayed have been based on the parameters previously found from the (uniform) individual component tests presented and debated in detail in Chapter 4.

From the performance graphs in Appendix B.3.3, one can see at which point the combined TMCSU segmentation results are let through the overshoot exclusion filter, noticeable as the first major step in the foreground percentages calculated from the combined binary foreground images. Prior to this point, only the LSPD and single differencing methods were operational, while the TMCSU background model was largely uninitialized, providing a segmentation result with $> 70\%$ foreground (70% was the limit set for the overshoot exclusion in Section 4.9). Furthermore, it can be seen how the TMCSU background estimates gradually become more accurate by inspecting the lowering foreground percentages; although for C3, the swift scene transitions repeatedly forces the background estimate to re-adapt, since not all background scenes in C3 are equal in terms of brightness and color composition, as seen in Figure B.3. However, the general fact that the scene transitions in C3 causes a major increase in the amount of foreground pixels indicates that the overshoot exclusion algorithm not manages to filter oversegmentation properly, maybe due to an inaccurate tuning parameter or that the method is too simple. Although the initialization period of the TMCSU model was not explicitly measured with the background initialization frame indicator (Section 3.8.3) in the simulations of the final system, the percentile amount of foreground pixels in the performance graphs in Appendix B.3.3 seems to stabilize after approximately 300 frames for C1, 100 frames for C2, and slightly less than 100 frames for C3. At these points, the buffers can roughly be considered initialized.

In Figure 5.1, coverage of the update blocking filter for C2, as well as the state of background estimate and the resulting combined TMCSU segmentation during buffer initialization is displayed. Apart from a portion of the fish, the update blocking filter covers most foreground structures accurately, which is partially reflected by the pattern of the uninitialized pixels in the background estimate. Furthermore, in can be seen how the uninitialized pixels from several channels propagate to the combined TMCSU result, which mostly only contains regions of erroneously segmented background. Similar responses can be seen after major scene transitions, where several background estimates have to be reevaluated, giving heavily oversegmented individual binary images. During such transitions, the overshoot exclusion appears to block the most severe oversegmentations, equal to the initial behavior for each simulation, where only the LSPD and single differencing methods were operational.

The combined effect of the S&KB-TMCSU update scheme and background smoothing can be seen in Figure B.8, where the TMCSU background model was stabilized and well defined. The estimates are homogeneous, thoroughly smooth and display no visible pollution from foreground structures, while the light gradients match that of the true background scene very well. In Section 3.7.1, it was discussed as to whether the combination of motion and edge data would manage to avoid smooth semi-stationary foreground objects, such as the artificial algae growth in C3, from being embedded into TMCSU background estimate. As can be seen in Figure 5.2, this scenario is not always handled by the TMCSU model update.

The individual contribution of the LSPD, single differencing and TMCSU segmentation modules to the total foreground binary image outputted by the background segmentation process can be seen in Figure 5.3, while several combined examples are illustrated in Figure 5.4, for C1 and C2, and Figure 5.5, for C3. One can clearly see how none of the segmentation methods singlehandedly manages to produce a comprehensive binary foreground image in Figure 5.3, and how combining the three drastically increases the consistency of the segmentation. From Figure 5.4, the increased consistency becomes visible in how the combined system seemingly performs equally well in most situations. The binary foreground images for C3 in Figure 5.5, however, are of less consistent quality, most likely caused by the restrictive, static binarization thresholds



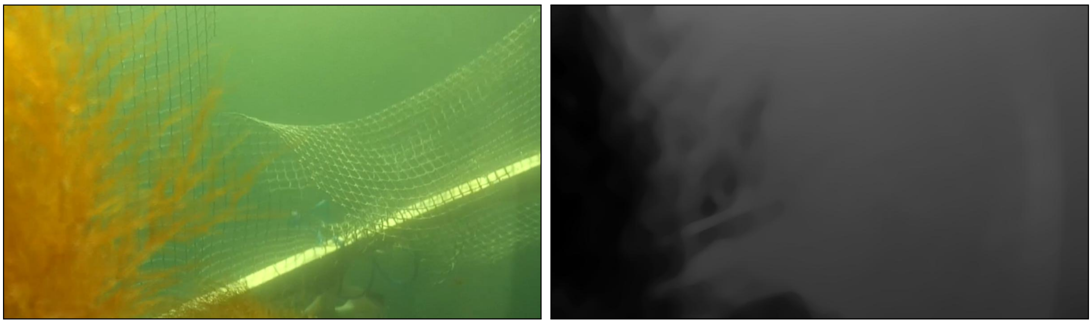
(a) Original frame

(b) Blue background Estimate

(c) Update blocking filter

(d) Combined TMCSU binary result

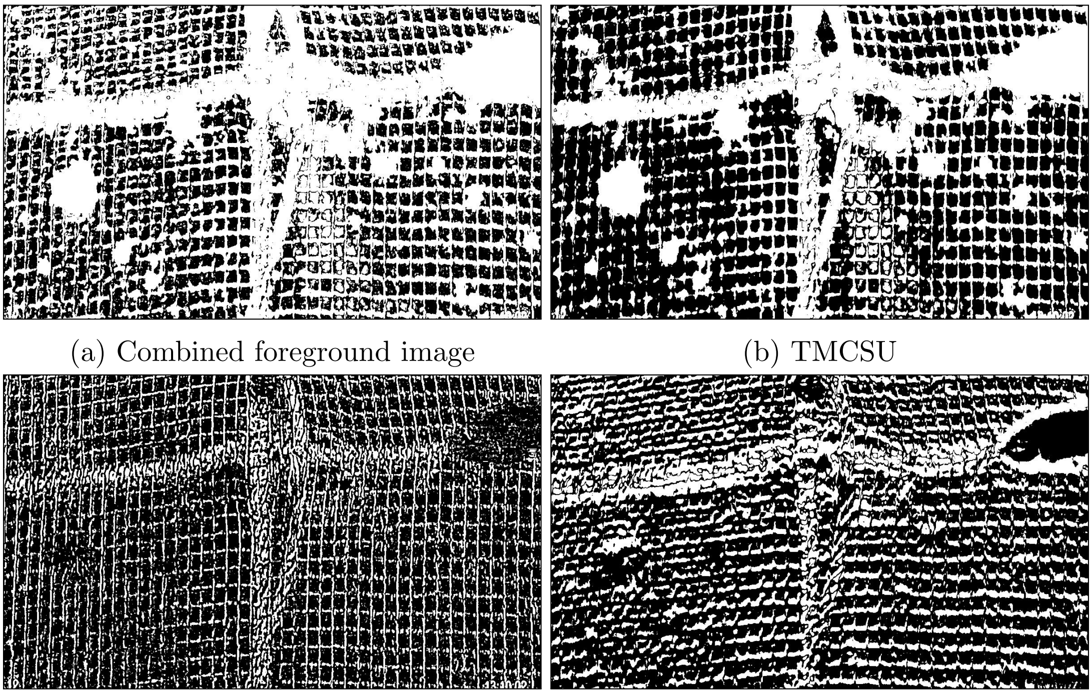
Figure 5.1: The state of various TMCSU components during buffer initialization for the C2, $k = 80$



(a) C3, $k = 520$

(b) Blue background Estimate

Figure 5.2: Stationary algae growth will be included into the TMCSU background estimate

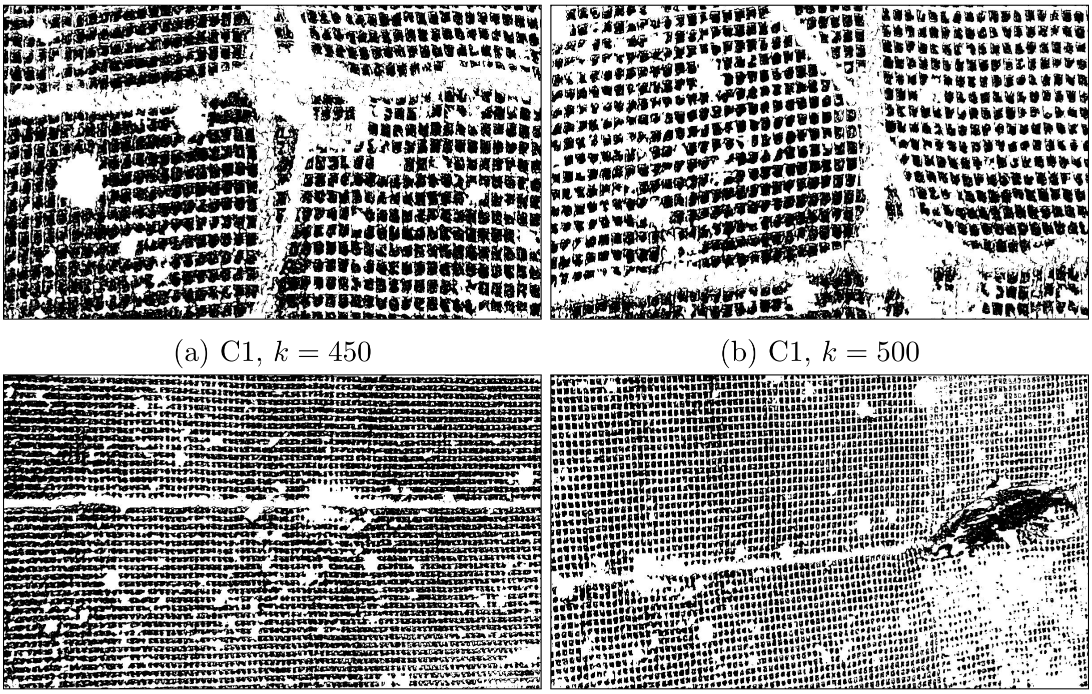


(a) Combined foreground image (b) TMCSU

(c) LSPD (d) Single Differencing

Figure 5.3: Composition of the binary foreground image outputted from the background segmentation process with a uniform combinatorial design, C1 $k = 320$ (Figure B.1c)

utilized for the uniform parameter scheme which for C3 showed undersegmentation in several cases (Section 4.6), but also the frequent scene transitions characteristic for C3 and the occurrences of stationary algae growth. Fish seemed to be the least reliably segmented foreground object in the analyzed videos, due to the extreme brightness variations from light reflections in its scales. Furthermore, when exposed to blur, the resulting lack of distinct image information reduced the segmentation consistency, as displayed for motion blur in Figure 5.4c, and optical lens blur in Figure 5.5g. In general, the TMCSU module displayed superior ability to segment algae growth, while the LSPD module showed the better performance in segmenting net threads; the single differencing module, on the other hand, displayed no particular strengths, but consistently provided partial segmentations of most objects with heightened performance in scenes with moderate motion. Also, some slight noise and erroneous segmentation of background pixels can be seen in the segmentation results.


(a) C1, $k = 450$

(b) C1, $k = 500$

(c) C2, $k = 300$

(d) C2, $k = 350$

Figure 5.4: Binary foreground image samples from the final system design

5.2 Damage Detection

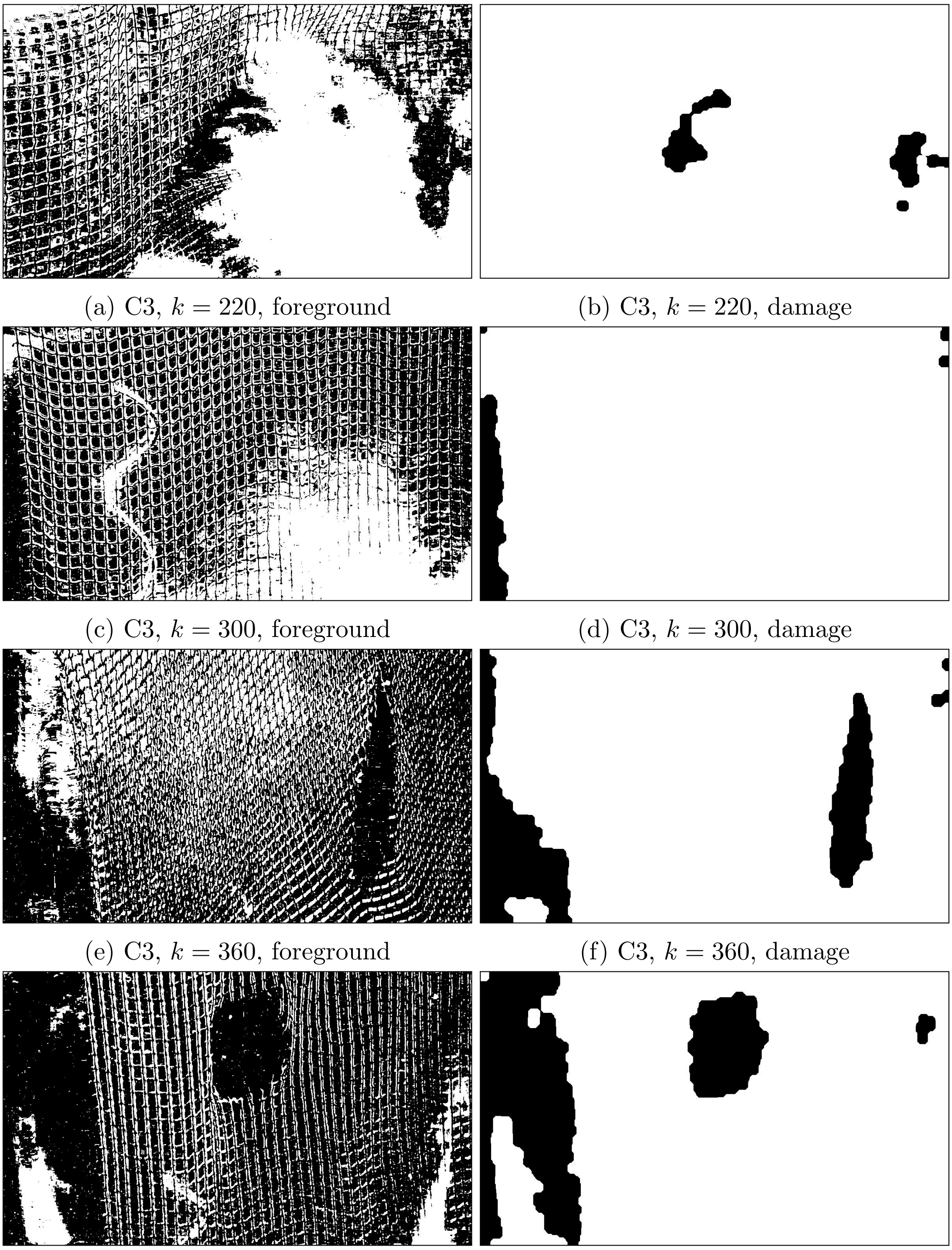
The "Damage Assessment" method in the final system is represented by the damage detection module in Figure 3.18. This module produces a binary image mask where black pixels indicate the lack of foreground structures, and therefore potentially also net damage. It requires some binary foreground image as input, which in the final system is provided by the background segmentation process described in the previous section. Since neither the C1 nor the C2 video material contain any knowledgeable net damage, only the binary foreground images from C3 will be analyzed.

In Table 5.1, the parameter selections utilized for the damage detection module has been listed. In this selection, the template size, T_s of the median filter was chosen as to best remove image noise while preserving vital details in the binary image. The radii of the morphological operations were selected large enough to properly exclude most undamaged regions, while not closing the regions actually damaged.

Median Filter, T_s	3
Closing Radius	14
Opening Radius	14

Table 5.1: Damage detection parameters

The binary damage images, or masks, from analyzing four different scenes in the C3 video material can be seen in Figure 5.5. In the binary foreground images, there are various instances of severe segmentation noise, isolated spots, sparse segmentation, erroneous segmentation of background pixels, as well as damages of various shapes and sizes, neither of which are surrounded by completely intact net structure; in short, they far from ideal. However, when investigating the resulting binary damage images, neither appear to be degraded in any particular manner. In most cases, the size of the detected damages are of closely equal size to the actual damaged regions, with smooth contours and regional consistency. The isolated spots in Figure 5.5a seems to somewhat reduce the regional fit of the detection, while sparsely segmented net areas in all images are misclassified. Actually, the only factor that appears to impact the damage mask, is the regional density of foreground pixels: if the density is sufficiently high, it will be classified as foreground, while sparsely segmented regions overall are classified as damage.



(a) C3, $k = 220$, foreground

(b) C3, $k = 220$, damage

(c) C3, $k = 300$, foreground

(d) C3, $k = 300$, damage

(e) C3, $k = 360$, foreground

(f) C3, $k = 360$, damage

(g) C3, $k = 450$, foreground

(h) C3, $k = 450$, damage

Figure 5.5: Results from damage detection algorithm for C3 with final system

Chapter 6

Discussion

In this chapter, the prospect, application and performance of the final system developed in this thesis will be discussed. Firstly, the background segmentation process and damage detection modules of the final system will be evaluated, in Section 6.1 and 6.2, respectively. Secondly, the prospect of the final system in terms of practical usability and fulfillment of the task of this thesis will be considered in Section 6.3. Finally, in Section 6.4, further courses of research will be suggested.

The LSPD and single differencing modules in the uniform combinatorial design scheme employed by the final system has already been covered in detail in Chapter 4. For this reason, only their main points and external behavior when combined with the TMCSU background model in the uniform design scheme will be evaluated in this chapter. The TMCSU background model and the uniform design scheme, on the other hand, has not been discussed previously, and will be covered in this chapter. Also, the various assumptions, methodologies and questions raised throughout this thesis will in this chapter be attempted answered.

6.1 Background Segmentation Process

One of the main focuses of this thesis has been to develop of a background segmentation scheme that provides a binary foreground image further to be used for net damage assessment. The final scheme incorporates a temporal background segmentation technique, a motion estimator and an edge detector. Ensuring stability towards unexpected foreign elements and scene changes has been the main goal in the design of the background segmentation process, where redundancy and module cooperation have been responses to this intention, as reflected by the uniform combinatorial design scheme. It was particularly noted in Section 1.3 how a temporal background segmentation technique would be adapted in order to work with a moving camera platform by assuming the background scene was static; TMCSU background model is the result of this task.

6.1.1 Combinatorial Design Methodology

In Section 4.1, it was found that different image channels consistently distinguished various image details better than others. This was confirmed when evaluating the unique parameter schemes and component designs for the LSPD, single differencing and temporal background segmentation techniques individually in Section 4.3, 4.4 and 4.6, respectively. With these findings in mind, the first guiding argument (Table 3.5) of the unique and uniform combinatorial system designs developed in Section 3.9 appears to hold true.

By analyzing the individual component performances in Chapter 4, as also excellently illustrated for the final system in Figure 5.3, the LSPD, single differencing and temporal background segmentation techniques employed in the combined background segmentation systems each seemed to hold a set of particular strengths and shortcomings in terms of which type of foreground elements they best segmented from the background scene. While all three segmentation modules typically managed to identify the major foreground elements in the scenes analyzed, the consistency of these segmentations varied for each method. Firstly, the LSPD method managed to segment sharp image points at a consistent rate, making it particularly suited for segmenting net threads, rope structure and other foreground elements of a well distinguished, non-smooth character. Algae growth, fish, and blurred image regions typically yielded a low response for the LSPD algorithm. Secondly, the TMCSU temporal segmentation method (as well as its TMSU and TMBU forerunners), thoroughly segmented algae growth and rope structure in a robust manner, but with a varying detection of fish and net structure. Lastly, the single differencing motion estimate showed a balanced performance, and managed to partially segment most foreground objects, as long as the relative motion between the foreground elements and the camera were non-zero. As such, it was found to distinguish net structure and fish on most occasions, but with an overall poor segmentation of smooth, weaving objects, such as algae growth and loose rope ends. From these observations, the second guiding argument (Table 3.5) of the unique and uniform combinatorial designs developed in Section 3.9 also appears to hold true.

The most concerning lack of stable segmentation of foreground elements provided by the combined segmentation systems evaluated, was their unreliability in terms of detecting fish. Due to the reflective scales of salmonids, the skin will appear both extremely bright, extremely dark, and everything in between, with quick succession. In addition, the fish skin also is largely untextured and smooth. Neither segmentation method seemed to handle this unpredictable appearance of fish robustly, as can be seen in Figure 5.4d for the final system. Since aquaculture sea cages rarely are inspected without holding fish, some secondary measure will have to be made in order to ensure that undetected fish avoid being classified as net damage. The typically rapid motion of fish might be a characteristic that potentially can be used for managing this issue, as will be discussed in the Further Work section of this chapter.

One of the main challenges, and also a possible limitation, of the combinatorial designs developed in this thesis, is the myriad of tuning parameters involved in each system's design. In a practical implementation of the final system, these settings will most likely require additional adjustment to work for different camera platforms, or be tuned to better fit specific scene types, or similar. Developing an automatic binarization method or parameter selection method might be worth consideration.

Comparison of Unique and Uniform Design Schemes

In Section 3.9.2, it was stated that if only segmenting distinct image points from multiple image channels and methods could manage to give a comprehensive combined segmentation result, then the uniform combinatorial method would also most likely be robust to scene changes, unlike the unique design scheme, and therefore outmatch the unique design scheme in most practical aspects. With a few exceptions, this was found to be true.

The unique component and parameter design schemes were found suboptimal for most tests conducted. Component experiments in Chapter 4 revealed that the unique parameter scheme was notoriously fragile to scene changes, since the DDT binarization with static parameters consistently failed at producing a binary result with a desirable segmentation amount when applied to any other video material than what the parameters were specifically tuned for. Furthermore, since each individual image channel rarely managed to produce a comprehensive segmentation result (as noted above), internal system components relying on the segmentation results from each other would often operate on suboptimal data, with a resulting overall degraded system performance. The unique design scheme did, however, show the potential segmentation accuracy from individual image channels that used finely tuned parameters, and therefore provided an indication of the possible benefit of finding a functional automatic binarization technique or developing an adaptive parameter scheme in a future system employing a uniform design.

The uniform design scheme, which was used during the final system tests of this thesis, showed overall stable and robust segmentation results. It was seen for the final system in Chapter 5 that the total redundancy from integrating multiple image channels and segmentation methods was capable of producing a comprehensive segmentation result, despite utilizing static binarization thresholds that only segmented the most distinct image points from each image channel. The beneficial effect of this redundancy was in particular notable by the overall quality of the segmentation results for C1 and C3, which for the single differencing and TMCSU methods used strongly compromising binarization parameters, as discussed in Section 4.4 and 4.6 respectively, due to the vast variations between the scenes in some of the video material. This compromise can be seen in Figure 5.4 and 5.5, where the net threads are inconsistently segmented in some regions, regions in which the LSPD edge detector was the primary contributor to the segmentation. Contrary to the TMCSU and single differencing techniques, the LSPD method performed at its best with a robust set of uniform parameters, as discussed in Section 4.3. From seeing how well the uniform design scheme performed even with suboptimal parameter settings, it seems likely that this combinatorial system design also may be applicable to real ROV inspection video feeds in the future, although possibly with a refined set of parameters.

Although the individual channel segmentation results for the unique parameter scheme did not manage to fully reproduce all foreground elements equally well, these segmentations were found more complete than those of each individual image channels with the uniform parameter scheme, both for the single differencing (Section 4.4) and temporal background segmentation (Section 4.6) methods. This might indicate that if the uniform parameter scheme had not utilized parameters which compromised the performance of these two methods, the performance for difficult scene elements, such as fish, might have been better. It is therefore reason to believe that the uniform design scheme would have performed even more consistently than its current state if the artificial net setup in C3 had not been included when deciding the uniform binarization parameters.

The overshoot exclusion method incorporated into the uniform design was intended to exclude segmentation results from individual image channels from the TMCSU method that potentially could cause oversegmentation in the combined binary foreground image outputted by the background segmentation process in the final system. In practice, this method did not avoid oversegmentation, as can be seen in the performance graphs of the final system in Appendix B.3.3. However, if oversegmentation due to an outdated TMCSU background model is found to be an issue in future implementations of the final system, a feature similar to the overshoot exclusion method might be desirable. Such a feature could for instance utilize measurements from the frame quality indicators for improved detection of individual channel oversegmentation.

6.1.2 Temporal Background Segmentation

A set of general evaluation criteria for the temporal background segmentation techniques in this thesis was listed in Table 3.4, which now will be used for evaluating the TMCSU background model in the final system.

The TMCSU background model displayed a drastic improvement of performance compared to its forerunners, the TMBU (Section 4.5) and TMSU (Section 4.8) models, as was seen for the final system results in Chapter 5. It appears that combining edge and motion information in the S&KB-TMCSU update scheme significantly heightened the consistency and accuracy of the update blocking filter, as can be seen for C2 in Figure 5.1. The resulting background estimates were smooth, homogeneous, and contained a low pollution of foreground pixels once fully initialized, as seen in Figure B.8. Actually, the background estimates found with the TMCSU model even display a significant improvement over the presumingly “ideal” background estimates found with the TMBU method (Appendix B.2.1), albeit using a buffer ten times smaller than that of the TMBU model. From this observation, one may conclude that the S&KB-TMCSU update scheme’s selective property functioned as desired.

In the discussion for the TMBU model tests in Section 4.5, the question as to whether the amount of foreground elements in the scene would affect the initialization period of the S&KB-TMCSU update scheme was raised. During the result analysis of the final system in Chapter 5, it was found that C1 initialized approximately three times slower than C2. When comparing the performance graphs from C1 and C2 (Appendix B.3.3), the amount of foreground content seems about equal, but the behavior of the motion graphs are different. It was described in Section 3.2 how C1 displays a slowly moving scene with recurring object positions, thereby explaining the dips in the motion plot of C1, whereas C2 contains video of an ROV traversing the net. It therefore seems likely that the amount motion present in the scene directly affects the initialization period of the TMCSU background model. With low amounts of motion and recurring object positions in the video, the update filter will constantly block the update of certain regions of the background estimate, prolonging the initialization period. Due to the small buffer used with the TMCSU model, it can potentially initialize rapidly, as was seen for C3, while the real initialization period seems to depend on the scene analyzed. The exact same deduction can be made for the learning rate of the TMCSU model, except the background estimate would need to adapt, or re-initialize, rather than initialize.

It was found from the TMBU component tests in Section 4.5 that decreasing the sampling interval directly would improve the responsiveness of the system while augmenting both the positive and the negative characteristics of the background model, and in addition increase the processing power required. Despite utilizing a small buffer and a low sampling interval, the TMCSU model appeared to robustly eliminate foreground objects from the background estimate while maintaining a low memory consumption and a high responsiveness to scene changes and events. The actual real-time capability of the TMCSU model, however, cannot be decided from these tests, but since it presumably can be calculated in linear time, and also features low memory requirements, it seems likely that it may be real-time applicable.

A potential flaw with the methodology of TMCSU model was described in Section 3.7.1 in terms of how stationary smooth foreground objects, such as the artificial algae growth in C3, might erroneously be included into the background estimate (Figure 5.2), resulting in erroneous segmentation of background pixels in subsequent frames (Figure 5.5c). This pollution seemed to happen consistently for the TMCSU model when evaluating the C3 video material. In order to avoid the deadlock scenario described in Section 2.3.2, one might not want to exclude smooth stationary foreground objects from the background estimate since the background itself is stationary and smooth. Excluding the stationary algae growth in C3 from the background update would therefore most likely also exclude the background itself. Furthermore, since algae growth typically does not remain stationary for longer periods of time during real ROV operations, this will most likely not be an issue in practice.

At several points throughout this thesis, it has been mentioned how the temporal buffer based background segmentation approaches investigated - the TMBU, TMSU and TMCSU methods - would not suit the rapidly changing scene conditions of the artificial C3 video material. Indeed, the flickering light reflections, or halos, unstable light conditions and stationary algae growth all caused erroneous segmentation of background pixels in the final segmentation results, as seen in Figure 5.5. However, since the additional challenges found in C3 were not found in any of the real ROV video material (C1 and C2), the occasional oversegmentation of the TMCSU method when applied to C3 will most likely not be an issue in a practical implementation of the final system. It should also be noted that even when exposed to the artificial light setting in C3, the TMCSU method, and the uniform design scheme as a whole, only displayed some minor cases of misclassified net damage, as seen in Figure 5.5, illustrating the general robustness of the system.

6.2 Damage Assessment

The main intention of the damage detection algorithm, has been to allow for a robust damage assessment that is less dependent on a comprehensive background segmentation result where all binary foreground structures are perfectly intact. Also, the method was to be designed so that the user of the system easily could define the minimum size of the damage to be detected.

In Section 5.2, it was seen that neither incomplete foreground structures, noise, nor the shape of the damages seemed obstruct the algorithm from providing a well defined, clean and accurate binary mask that fit the regional size of the damages well, as seen in Figure 5.5. The general observation was that only the regional density of foreground pixels appeared to influence the outcome of the algorithm, as was verified by the algorithm's ability to avoid misclassification even in image regions where only few foreground points were available. When comparing the calculated binary masks to the real damages in the original frames in Figure B.3, one will see that the fit is not always accurate; however, the damage detection algorithm consistently provided highly favorable results given the quality of the binary foreground images it analyzed, and the lack of segmentation accuracy in the binary foreground images rather questions the operation of the background segmentation process. As such, the damage detection algorithm presented in this thesis seems to suit the aim of achieving robust damage assessment in an excellent manner.

The size of the damages detected is controlled by the radius of the disk-shaped structuring element used for the morphological operations incorporated into the algorithm. Since the size of the structuring element directly affects the size of the damages to be detected, it seems like an intuitive and generally well suited parameter for the user to access during ROV inspections. However, since the actual radius is measured in pixels, the size of the damages detected depends on the distance between the camera and the net wall. In order to achieve a user-friendly system, the radius should therefore be coupled with some distance measurement to allow for automatic scaling of the disk radius used. An accurate measure of net distance was found by (Jakobsen, 2011), and adjusting the size of the structuring element would presumingly be a simple linear scaling operation.

The potential accuracy of the damage detection system is entirely dependent on the quality of the binary foreground image from the background segmentation process. With the camera resolution and segmentation consistency seen in this thesis, finding single mesh damages seems unlikely. However, there appears to be no theoretical limitations in the methodology and algorithmic choices made in the final system that would restrain the detection accuracy in a future system.

Seeing the clear division between the damaged and non-damaged regions in the binary damage image, and the accurate fit of these regions to true size of the damages, using the binary mask as a direct overlay for the ROV operators video feed seems feasible. Furthermore, calculating some center position of each "blob" should be very well possible, allowing the results to be utilized as a reference point for an eventual path planning algorithm. A few practical approaches for solving these tasks, were mentioned in Section 3.10.

6.3 Prospect of the Final System

In the task description that motivated this thesis, a methodology for robust, automatic detection of net damages based on camera vision, with applicability to automatic positioning and path planning in an ROV operation, was requested. The aim of this request was to increase the regularity and efficiency of ROV operations by alleviating the responsibilities of the human operator. In the introduction, this was further elaborated by the aim of providing data for a visual tool that would assist the ROV operator during routine net inspections. Within some practical limits, the system developed in this thesis seems to largely fulfill these requests.

In its current state, the background segmentation process managed to accurately segment most foreground elements in a robust manner. Considering how the system has performed so far, in particular for C3, it is also expected to function for video material not analyzed in this thesis. The irregular segmentation of fish remains one of the largest concerns with this design, although this might be less influential if the system is tuned to robustly tolerate realistic video material only. The damage detection system proved robust to difficult and inconsistent segmentation results and identified all damaged regions. However, it also yielded false positive detections in the situations where the background segmentation process was incapable of providing even a sparse reproduction of foreground details, an issue that presumingly might be problematic with fish. Despite the superior performance of the damage detection method itself, the system can therefore not be deemed entirely robust at this point. Since this dilemma could occur for foreign objects not analyzed as well, a practical implementation might require additional measures to identify true from false positive detections. Also, since no distance measurement was employed in this thesis, a suitable relation between the user-definable size of damages detected and the distance to the net will have to be found.

The binary damage images produced by the system shows potential for direct further usage, either as a video stream overlay to assist an ROV operator, or in combination with automatic positioning and path planning in some future system. While the center point of the damages detected could work well as a reference point for an automatic navigational system, the overlay might help the ROV operator detect damages otherwise overlooked. If the system also notified the ROV operator each time something suspicious is found, not only would that allow the operator to make a closer investigation of the irregularity detected and highlighted, but also make the inspector more attentive by breaking up the otherwise monotone inspection routine. As an assisting system for manual ROV inspections, occasional false positive detections from fish would also be easily dismissed by the ROV operator, which would recognize such error immediately. For an automatic system application, however, erroneous detections from fish could potentially confuse the system, and therefore need some measure of detection validation.

6.4 Further Work

The final system proposed in this thesis did not robustly distinguish moving fish from the background, potentially resulting in frequent misclassification of net damage if implemented in practice without secondary measures. The fish appears to normally move significantly faster than the remaining foreground structures in the video streams, and might therefore be separable by their overall high relative motion to other foreground elements. One mean of doing so could be to implement a traditional optical flow algorithm that not only detects motion, but also calculates a optical flow field with measurable field vectors, and then exclude regions of significant motion. Another measure could be to track the movement of the entire binary foreground image, for instance with binary image correlation, and effectively low-pass filter all values to be included in the final binary foreground image, where a foreground point only would be included in the final binary result if it had appeared in the same relative position in the detected foreground image over some number of previous frames. Such a binary foreground evaluation scheme could most likely eliminate spontaneous changes in the detected foreground in general, either occurring from passing fish, noise, sporadic lightning scene changes, foreign objects or similar, as well as solidify the foreground image in general.

One of the main challenges of this thesis was to overcome the lack of an automatic, global binarization method for implementation with the single differencing and temporal background segmentation methods of this thesis. A possible manner of achieving such an automatic binarization could be to compared the percentile amount of sharp pixels to the foreground segmentation amount, and adjust the DDT parameters thereafter. This is based in the observation that the sharp pixel count seemed relative, but stable, for each scene, while the foreground segmentation typically followed the plots of the motion and sharpness amounts when operating well.

The final system from this thesis incorporates a myriad of tuning parameters that must work together for a proper result. In addition, interactions, relationships and the final effects of different parameter combinations can be difficult to analyze in complex combinatorial design schemes. An advanced approach to automatically find a suitable set of parameters for the final system, could be to utilize an evolutionary algorithm for parameter testing. A well proven, real time applicable evolutionary algorithm, such as the Quantum-inspired Evolutionary Algorithm proposed by Kuk-Hyun Han and Jong-Hwan Kim might be a viable option (Han & Kim, 2000)(Han & Kim, 2002).

Several aspects of the final system could potentially be improved by incorporating information from the ROV's sensory equipment. In order to avoid smooth foreground objects in polluting the background estimate of the TMCSU model when the ROV is stationary, the S&KB-TMCSU update scheme could incorporate sensor data from the ROV's navigation system directly, and block the update of the background model when no motion is registered. Also, the size of the structuring element for the damage detection could be coupled with a distance measuring tool if employed by the ROV platform.

If no distance measuring tool is incorporated into the ROV platform, the net distance might be calculated by analyzing the Fourier transforms of subdivisions of the current video frame. Due to the linear relation between rotation and scale of the patterns generated with the Fourier transform when analyzing repetitive textures (Nixon & Aguado, 2002), one might be able to find a relation between the Fourier transform of net segments and their distance from the camera. However, not only could this provide a distance measurement, it also could be used to calculate the physical orientation of the ROV relative to the net wall. A similar calculation of the relative orientation between the camera platform and scene objects was made with the purpose of image rectification based on roll, pitch and yaw angles in (Haugene, 2013).

The overshoot exclusion algorithm in this thesis did not add any particular benefits to the final system. It could, however, be useful to do separate combining operations of each segmentation module instead of using a basic OR operator. A more advanced overshoot exclusion mechanism that adjusts to each scene and image channel might be desirable. Such a module might for instance be based on the frame quality indicators in Section 3.8. To automatically reset the background model after consistent channel overshoot might be another possible solution.

Chapter 7

Conclusion

Accompanying the growth of the aquaculture fish farming industry in the recent years, the usage of ROVs for routine net inspections has become increasingly common, a task which for a human operator can be monotonous and time consuming. In order to increase the efficiency and regularity of ROV net inspections, realizing an automatic net damage assessment system is of interest. The purpose of this study was therefore to develop a robust methodology for assessing net damage with an ROV mounted camera and computer vision. The findings clearly suggest that utilizing a high-redundancy combinatorial design for background segmentation, followed by a damage assessment technique based on morphological operations, can be a viable approach for achieving a robust evaluation of net damage during underwater net inspection. It was seen that by combining the segmentation results from multiple techniques and image channels, the shortcomings of each individual component was complemented by the strengths of the others, giving a robust, and mostly accurate, segmentation of the background. Fish was particularly difficult to segment reliably due to its unpredictable appearance, and could cause false positive detections of damage if not handled properly in a practical implementation. The damage detection scheme produced a clear and accurate binary damage mask that covered the regions of the true net damage well, as long as the background segmentation result yielded a minimal, but complete, representation of all foreground structures. The basic nature of the calculated damage mask seems suited for usage with an automatic navigation system, or as a visual assistance for a human ROV operator, without major modifications. Further research is needed, but the methodology purposed can be recommended for a future implementation in a practical system.

References

Arbeidstilsynet. (2011). *Arbeidsmiljø og sikkerhet i havbruk.* Retrieved from http://www.arbeidstilsynet.no/brosjyre.html?tid=227913/

Bouwmans, T., El Baf, F., Vachon, B., et al. (2008). Background modeling using mixture of gaussians for foreground detection - a survey.

Bradski, G. (2000). Opencv library documentation. *Dr. Dobb's Journal of Software Tools*.

Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*(6), 679–698.

Carlsen, A. L. (2010). *Navigational assistance for mini-rov* (Unpublished master's thesis). Norwegian University of Science and Technology.

Cristani, M., Farenzena, M., Bloisi, D., & Murino, V. (2010, February). Background subtraction for automated multisensor surveillance: A comprehensive review. *EURASIP J. Adv. Signal Process*, *2010*, 43:1–43:24. Retrieved from http://dx.doi.org/10.1155/2010/343057 doi: 10.1155/2010/343057

Cucchiara, R., Grana, C., Piccardi, M., & Prati, A. (2000). Statistic and knowledge-based moving object detection in traffic scenes. In *Intelligent transportation systems, 2000. proceedings. 2000 ieee* (p. 27-32). doi: 10.1109/ITSC.2000.881013

Cucchiara, R., Grana, C., Piccardi, M., & Prati, A. (2003, Oct). Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *25*(10), 1337-1342. doi: 10.1109/TPAMI.2003.1233909

Cucchiara, R., & Piccardi, M. (1999). Vehicle detection under day and night illumination.

Efford, N. (2000). *Digital image processing: A practical introduction using java (with cd-rom)* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Elgammal, A. M., Harwood, D., & Davis, L. S. (2000). Non-parametric model for background subtraction. In *Proceedings of the 6th european conference on computer vision-part ii* (pp. 751–767). London, UK, UK: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=645314.649432

FAO. (2012). *The state of world fisheries and aquaculture - 2012.* Food and Agriculture Organization of the United Nations. Retrieved from http://www.fao.org/docrep/016/i2727e/i2727e.pdf

Fleming, I. A., & Einum, S. (1997). Experimental tests of genetic divergence of farmed from wild atlantic salmon due to domestication. *ICES Journal of Marine Science: Journal du Conseil*, *54*(6), 1051-1063. Retrieved from http://icesjms.oxfordjournals.org/content/54/6/1051.abstract doi: 10.1016/S1054-3139(97)80009-4

Han, K.-H., & Kim, J.-H. (2000). Genetic quantum algorithm and its application to combinatorial optimization problem. In *Evolutionary computation, 2000. proceedings of the 2000 congress on* (Vol. 2, p. 1354-1360 vol.2). doi: 10.1109/CEC.2000.870809

Han, K.-H., & Kim, J.-H. (2002, Dec). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *Evolutionary Computation, IEEE Transactions on*, *6*(6), 580-593. doi: 10.1109/TEVC.2002.804320

Haugene, T. (2013). *Analyzing the structural integrity of fish farms using an unmanned aerial vehicle and image geometry evaluation methods.*

Hayes-Roth, F., Waterman, D., & Lenat, D. (1983). *Building expert systems.* Addison-Wesley.

Heide, M. A., & Moe, H. (2004, October). *Alternative notkonsepter - delrapport i prosjekt ”nye rømningssikre merdkonsept”* (Tech. Rep.). SINTEF.

Hindar, K., Ryman, N., & Utter, F. (1991). Genetic effects of cultured fish on natural fish populations. *Canadian Journal of Fisheries and Aquatic Sciences*, *48*(5), 945-957. Retrieved from `http://www.nrcresearchpress.com/doi/abs/10.1139/f91-111` doi: 10.1139/f91-111

Jakobsen, R. A. H. (2011). *Automatic inspection of cage integrity with underwater vehicle* (Unpublished master's thesis). Norwegian University of Science and Technology.

Jensen, Ø., Dempster, T., Thorstad, E. B., Uglem, I., Fredheim, A., et al. (2010). Escapes of fishes from norwegian sea-cage aquaculture: Causes, consequences and prevention. *Aquaculture Environment Interactions*, *1*(1), 71–83.

Joblove, G. H., & Greenberg, D. (1978, August). Color spaces for computer graphics. *SIGGRAPH Comput. Graph.*, *12*(3), 20-25. Retrieved from `http://doi.acm.org/10.1145/965139.807362` doi: 10.1145/965139.807362

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Retrieved from `http://www.cs.unc.edu/ welch/kalman/media/pdf/Kalman1960.pdf`

Kameda, Y., & Minoh, M. (1996, January). A human motion estimation method using 3-successive video frames. *Proceedings of International Conference on Virtual Systems*
ldots. Retrieved from `http://www.imel1.kuis.kyoto-u.ac.jp/members/kameda/research/publication/1996/VSMM/vsmm96.ps`

Karaman, M., Goldmann, L., Yu, D., & Sikora, T. (2005). Comparison of static background segmentation methods. In (Vol. 5960, p. 596069-596069-12). Retrieved from `http://dx.doi.org/10.1117/12.633437` doi: 10.1117/12.633437

Koller, D., Weber, J., & Malik, J. (1993). Robust multiple car tracking with occlusion reasoning. In (pp. 189–196). Springer-Verlag.

Lo, B. P. L., & Velastin, S. (2001). Automatic congestion detection system for underground platforms. In *Intelligent multimedia, video and speech processing, 2001. proceedings of 2001 international symposium on* (p. 158-161). doi: 10.1109/ISIMP.2001.925356

McGinnity, P., Stone, C., Taggart, J. B., Cooke, D., Cotter, D., Hynes, R., … Ferguson, A. (1997). Genetic impact of escaped farmed atlantic salmon (salmo salar l.) on native populations: Use of dna profiling to assess freshwater performance of wild, farmed, and hybrid progeny in a natural river environment. *ICES Journal of Marine Science: Journal du Conseil*, *54*(6), 998-1008. Retrieved from http://icesjms.oxfordjournals.org/content/54/6/998.abstract doi: 10.1016/S1054-3139(97)80004-5

Nixon, M., & Aguado, A. S. (2002). *Feature extraction & image processing, first edition* (1nd ed.). Newnes.

NS9415. (2009). *Marine fish farms - requirements for site survey, risk analyses, design, dimensioning, production, installation and operation* (No. NS9415). Standard Norge.

OED. (2014, April). Oxford English Dictionary @ONLINE. Retrieved from www.oxforddictionaries.com

Olafsen, T., Winther, U., Olsen, Y., & Skjermo, J. (2012). *Versiskapning basert på produktive hav i 2050* (Tech. Rep.). DKNVS and NTVA. Retrieved from http://www.ntnu.no/documents/15827539/0/verdiskaping-basert-pa-produktive-hav-i-2050.pdf

Olsen, M. E. (2013). *Camera assisted rov navigation in sea cages* (Unpublished master's thesis). Norwegian University of Science and Technology.

Otsu, N. (1979, Jan). A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, *9*(1), 62-66. doi: 10.1109/TSMC.1979.4310076

Piccardi, M. (2004, October). Background subtraction techniques: A review. In *Systems, man and cybernetics, 2004 ieee international conference on* (Vol. 4, p. 3099-3104). doi: 10.1109/ICSMC.2004.1400815

Plataniotis, K. N., & Venetsanopoulos, A. N. (2000). *Color image processing and applications*. New York, NY, USA: Springer-Verlag New York, Inc.

Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, *13*(1), 146-168. Retrieved from http://dx.doi.org/10.1117/1.1631315 doi: 10.1117/1.1631315

Sletta, M. S. (2013). *Machine vision for real time detection of net damage in sea cages* (Unpublished master's thesis). Norwegian University of Science and Technology.

Sunkavalli, K., Joshi, N., Kang, S. B., Cohen, M., & Pfister, H. (2012, November). Video snapshots: Creating high-quality images from video clips. *Visualization and Computer Graphics, IEEE Transactions on*, *18*(11), 1868-1879. doi: 10.1109/TVCG.2012.72

Taylor, M., & Kelly, R. (2010). Assessment of protocols and development of best practice contingency guidance to improve stock containment at cage and land-based sites. *Scottish Aquaculture Research Forum (SARF)*, *1*. Retrieved from http://www.sarf.org.uk/cms-assets/documents/28923-913107.sarf054---volume1.pdf

Wedel, A., & Cremers, D. (2011). *Stereo scene flow for 3d motion analysis*. Springer. Retrieved from http://www.springer.com/978-0-85729-964-2

Appendix A

Terminology

A.1 Acronyms

NTNU	Norsk Teknisk Naturvitenskapelige Universitet (Norwegian University of Science and Technology)
SINTEF F&A	Selskapet for Industriell og Teknisk Forskning Fiskeri og Havbruk (The Society for Industrial and Technological Research Fisheries and Aquaculture)
ROV	Remotely Operated Vehicle
AUV	Autonomous Underwater Vehicle
CV	Computer Vision
INS	Inertial Navigation System
GUI	Graphical User Interface
DFS	Depth First Search
FG	Foreground
BG	Background
RGB	Red Green Blue
HSI	Hue Saturation Intensity
HSV	Hue Saturation Value
HLS	Hue Lightness Saturation
GMM	Gaussian Mixture Model
KDE	Kernel Density Estimation
RTSS	Real-time Traffic Surveillance System
S&KB	Statistical & Knowledge Based
FIFO	First In First Out

OF	Optical Flow
FPS	Frames Per Second
DDT	Double Direct Thresholding
LSPD	Local Sharpness Point Detector
TMBU	Temporal Median Model w/ Blind Update
TMSU	Temporal Median Model w/ S&KB Selective Update
TMCSU	Temporal Median Model w/ Modified S&KB Combinatorial Selective Update
LR	Learning Rate
IP	Initialization Period
SNR	Signal to Noise Ratio
C	Video clip (C1, C2 and C3)

A.2 Glossary

Image information format The conceptual meaning of relative gray tones in some grayscale image (e.g. an image and its inverted representation have different information formats).

Oversegmentation When a segmented image region contain more than the optimal amount of binary ones

Undersegmentation When a segmented image region contain less than the optimal amount of binary ones

Segmentation inversion When a background pixels are segmented as if they were foreground pixels, and vice versa.

Uniform Parameters Parameter settings tuned to function equally well for all scenes analyzed

Unique Parameters Parameter settings tuned specifically for each individual scenes analyzed

Ideal Background Estimate The smoothest and least polluted background estimate obtainable when analyzing a continuously congested scene

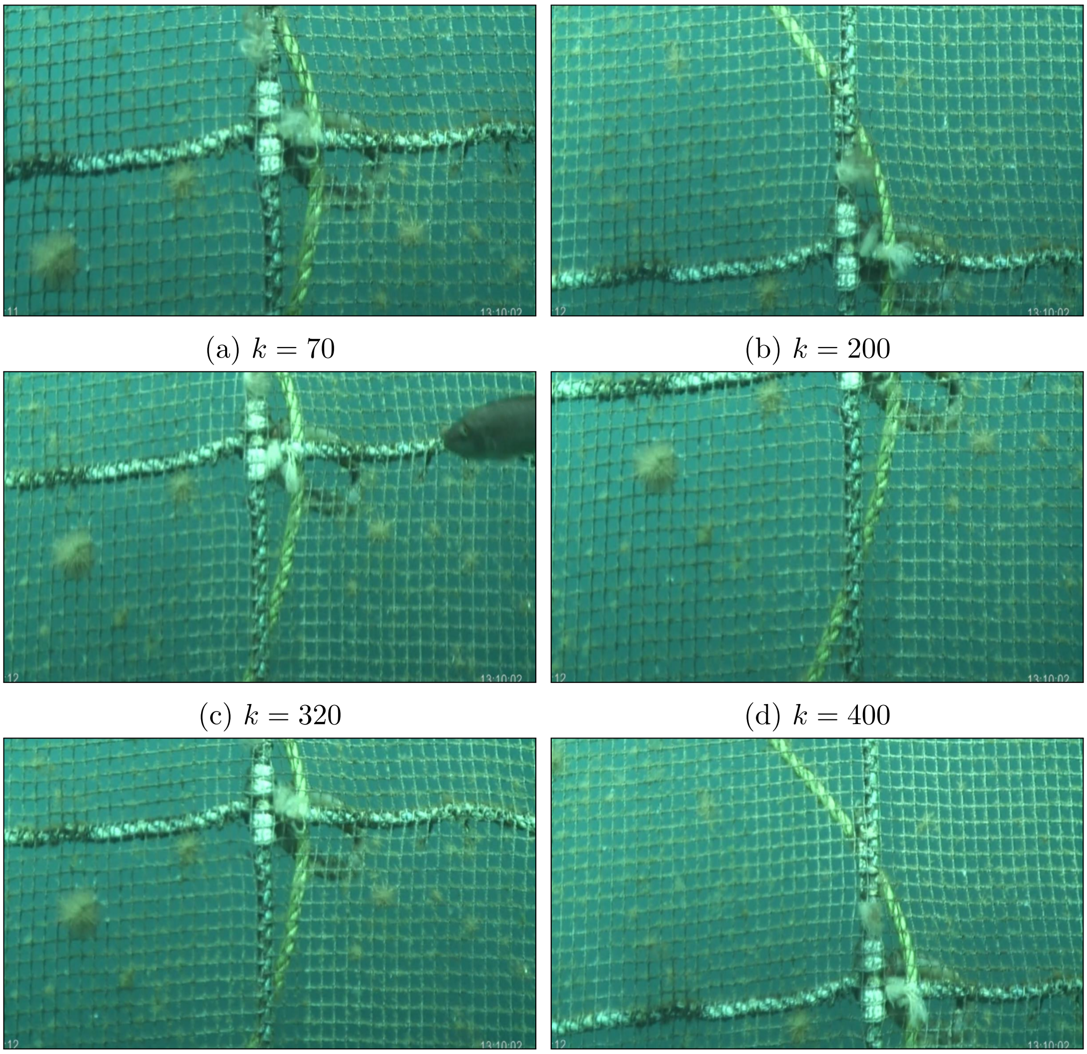
A.3 Nomenclature

t	time index
k	frame index
p	Image point
F_k	Binary foreground image at frame k
B_k	Background estimate at frame k
B_k^p	Background estimate point p at frame k value
I_k	Image at frame k
B_t	Background estimate at time t
I_t	Image at time t
σ	Standard deviation
G_σ	Gaussian filter
T_s	Template size
$\triangle t$	Sampling interval
m	Sampling Frequency
$U(\cdot)$	Placeholder function
n	Buffer size
w_b	Background preservation rate
T_f	Frame rate
OF^k	Calculated optical flow for some foreground point FG^k
TH	Optical flow threshold
$(x,\, y)$	pixel coordinates
$(u,\, v)$	positional deviation for some point (x, y)
W_k	Local sharpness indicator
T	Threshold
T_U	Upper threshold
T_L	Lower threshold
s	Structuring element
I_{Diff}	Difference image
F_k^p	Update blocking filter for point p and frame k
Q	Running mean buffer size
L	Percentile limit threshold

Appendix B

Reference Media

B.1 Test Video Clip Samples



(a) $k = 70$

(b) $k = 200$

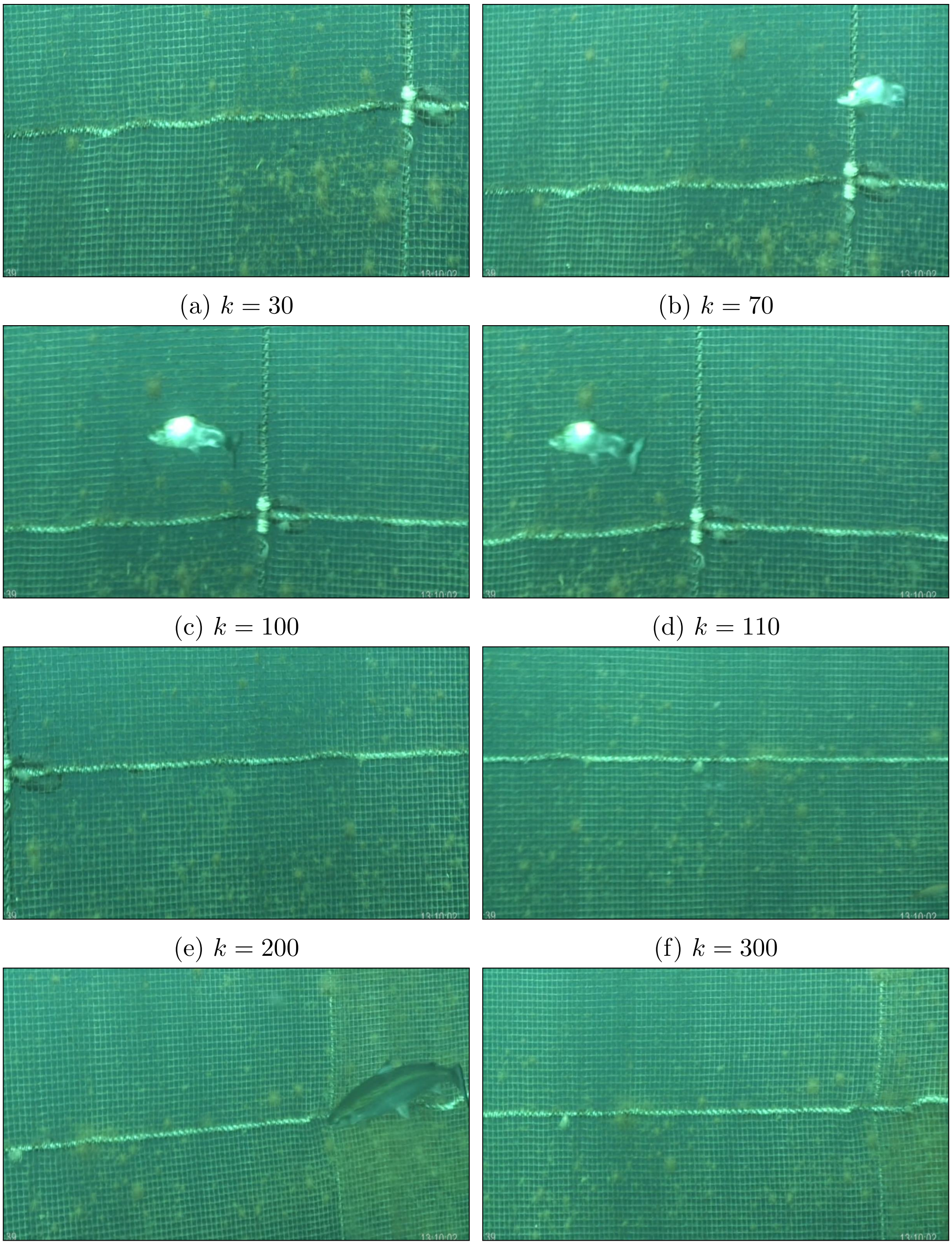
(c) $k = 320$

(d) $k = 400$

(e) $k = 450$

(f) $k = 500$

Figure B.1: Images from the C1 video compilation



(a) $k = 30$

(b) $k = 70$

(c) $k = 100$

(d) $k = 110$

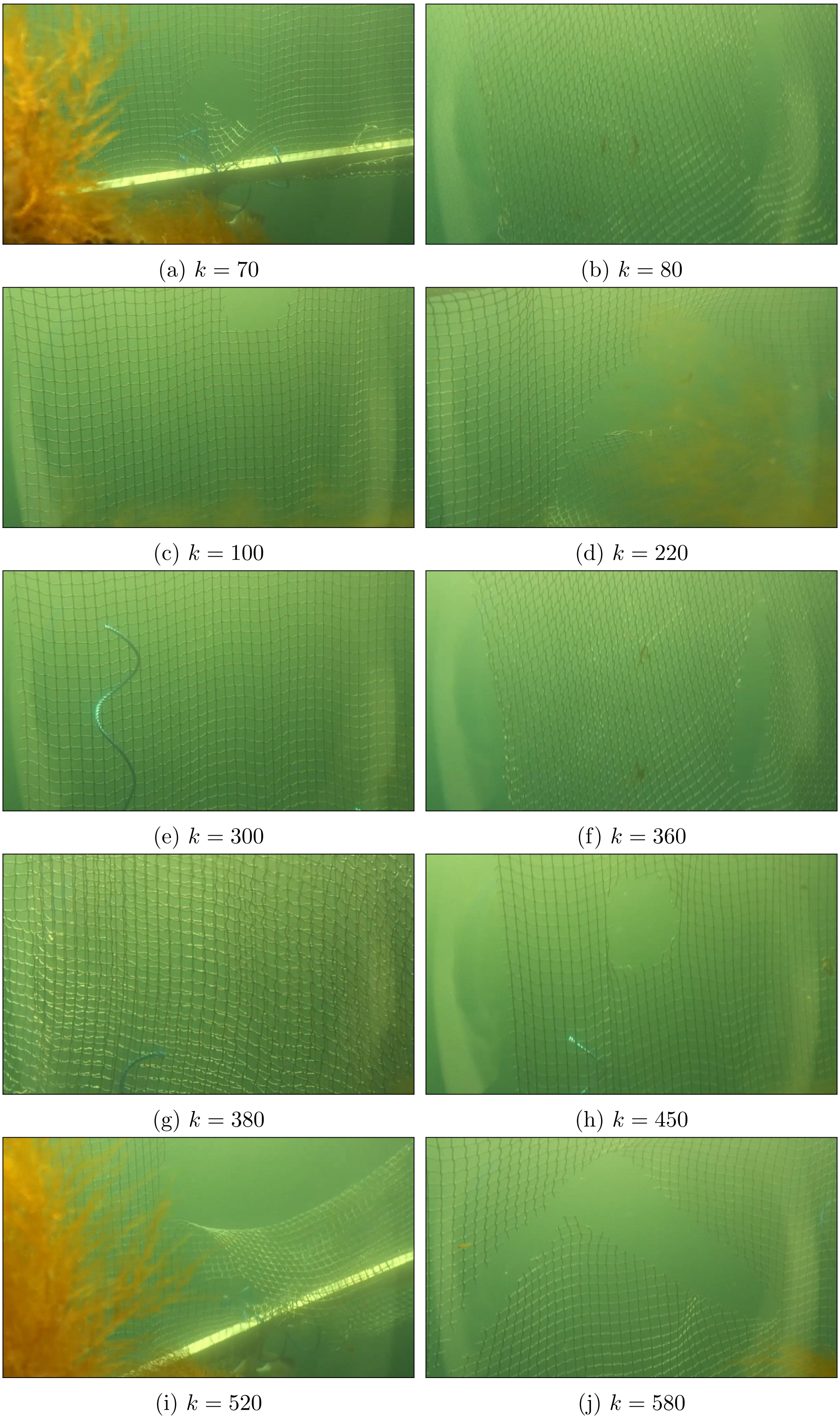
(e) $k = 200$

(f) $k = 300$

(g) $k = 350$

(h) $k = 400$

Figure B.2: Images from the C2 video compilation



(a) $k = 70$

(b) $k = 80$

(c) $k = 100$

(d) $k = 220$

(e) $k = 300$

(f) $k = 360$

(g) $k = 380$

(h) $k = 450$

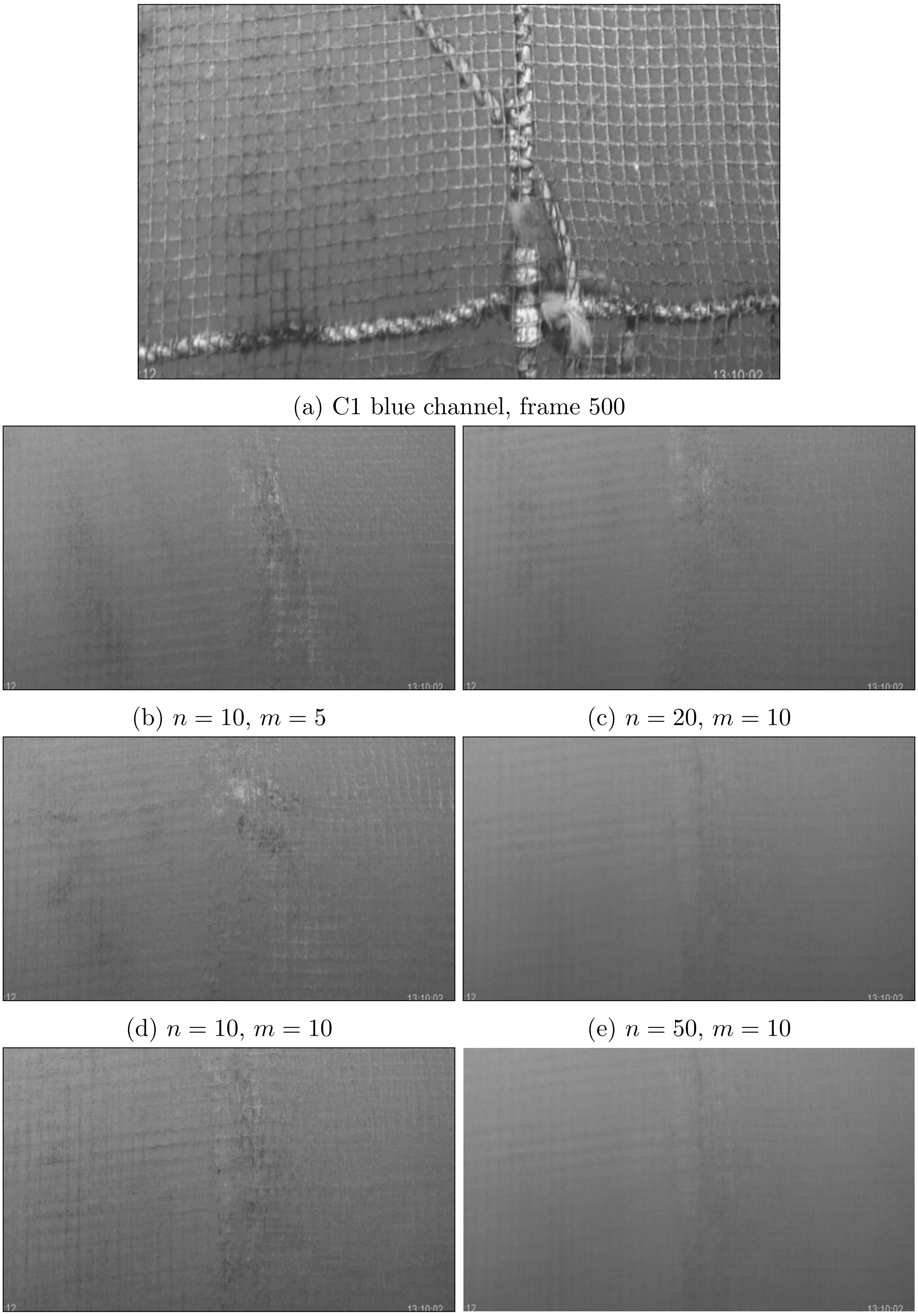
(i) $k = 520$

(j) $k = 580$

Figure B.3: Images from the C3 video compilation

B.2 Background Estimates

B.2.1 TMBU Background Estimates



(a) C1 blue channel, frame 500

(b) $n = 10$, $m = 5$

(c) $n = 20$, $m = 10$

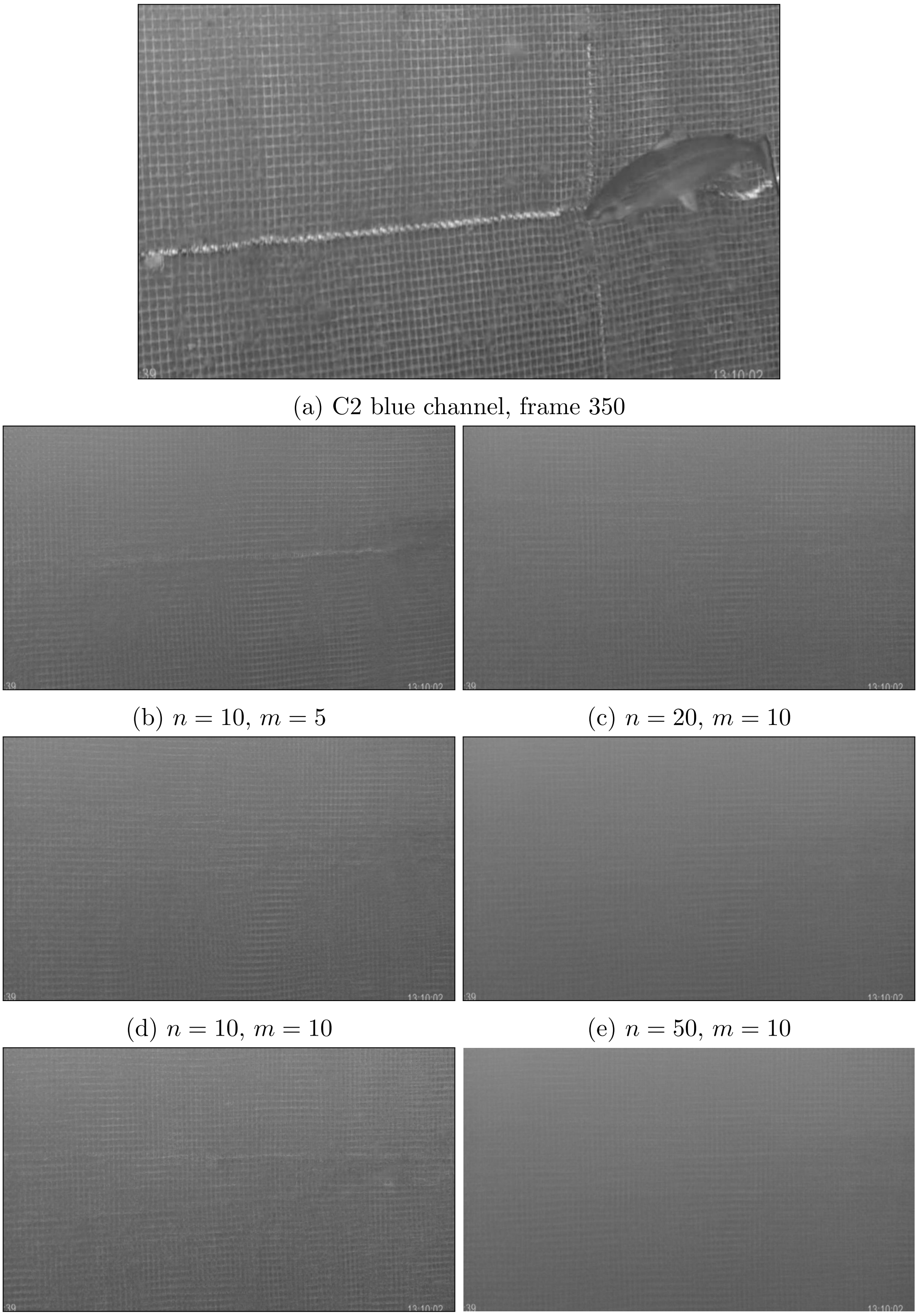
(d) $n = 10$, $m = 10$

(e) $n = 50$, $m = 10$

(f) $n = 10$, $m = 50$

(g) $n = 100$, $m = 10$

Figure B.4: TMBU BG estimates for various buffer settings, C1 blue channel



(a) C2 blue channel, frame 350

(b) $n = 10$, $m = 5$

(c) $n = 20$, $m = 10$

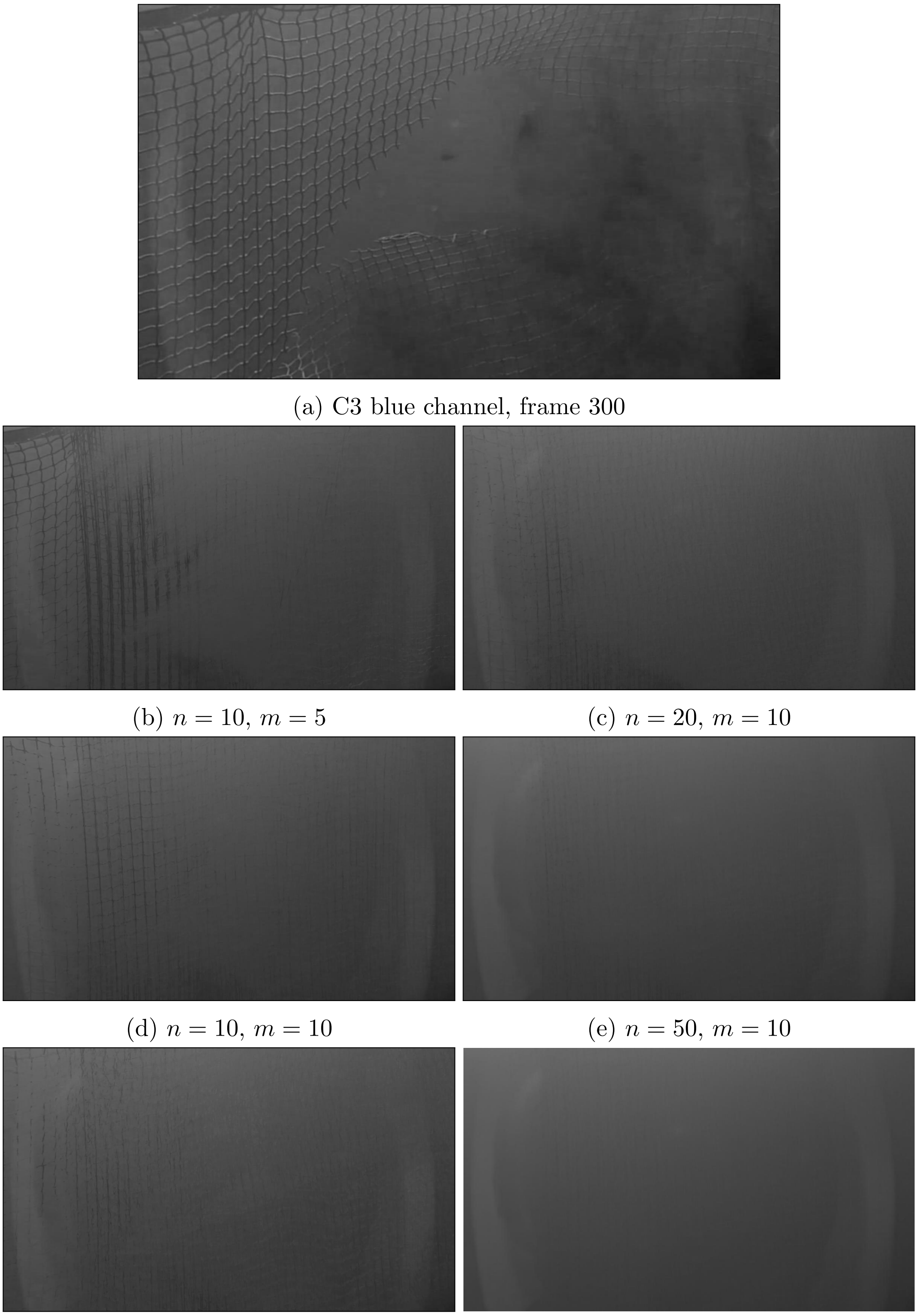
(d) $n = 10$, $m = 10$

(e) $n = 50$, $m = 10$

(f) $n = 10$, $m = 50$

(g) $n = 100$, $m = 10$

Figure B.5: TMBU BG estimates for various buffer settings, C2 blue channel



(a) C3 blue channel, frame 300

(b) $n = 10$, $m = 5$

(c) $n = 20$, $m = 10$

(d) $n = 10$, $m = 10$

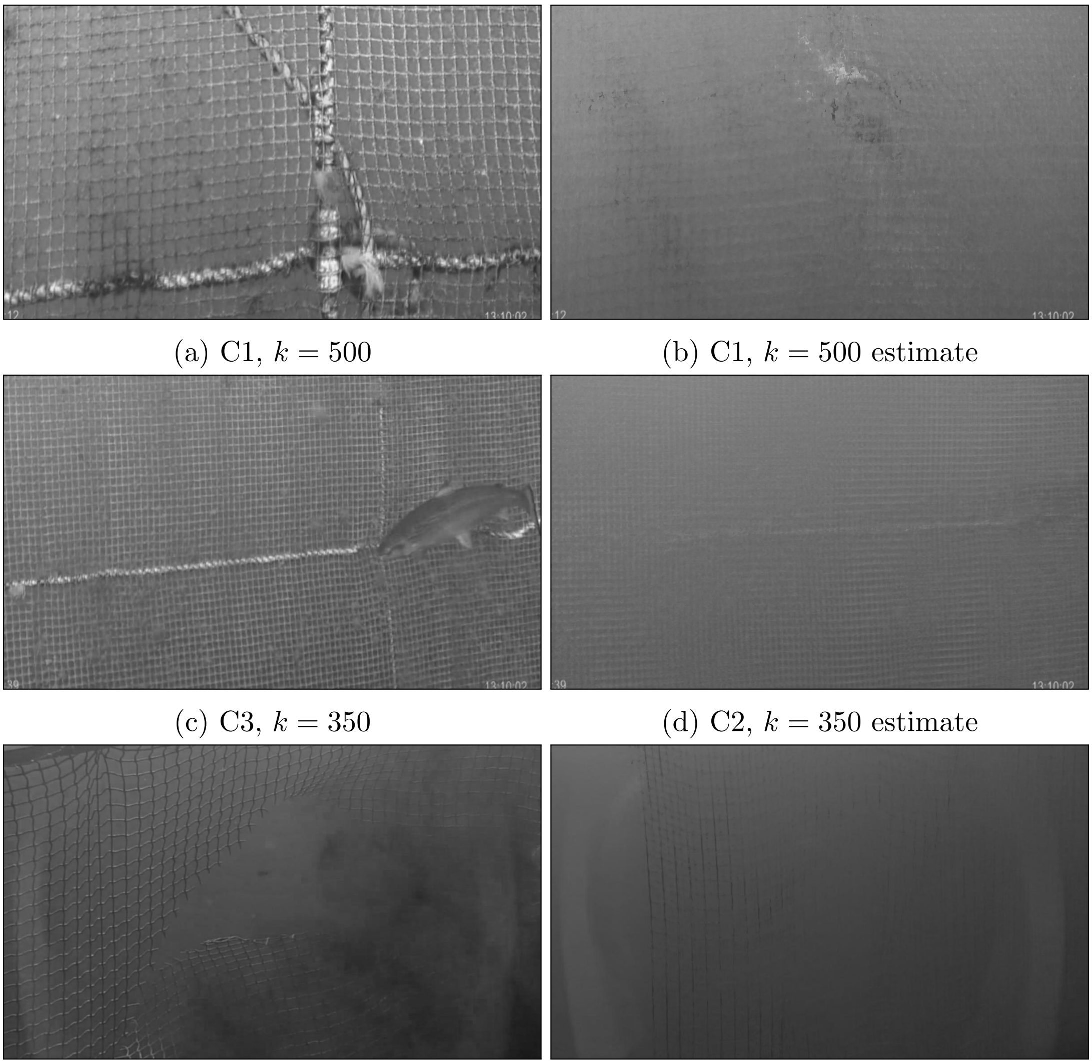
(e) $n = 50$, $m = 10$

(f) $n = 10$, $m = 50$

(g) $n = 100$, $m = 10$

Figure B.6: TMBU BG estimates for various buffer settings, C3 blue channel

B.2.2 TMSU Background Estimates



(a) C1, $k = 500$

(b) C1, $k = 500$ estimate

(c) C3, $k = 350$

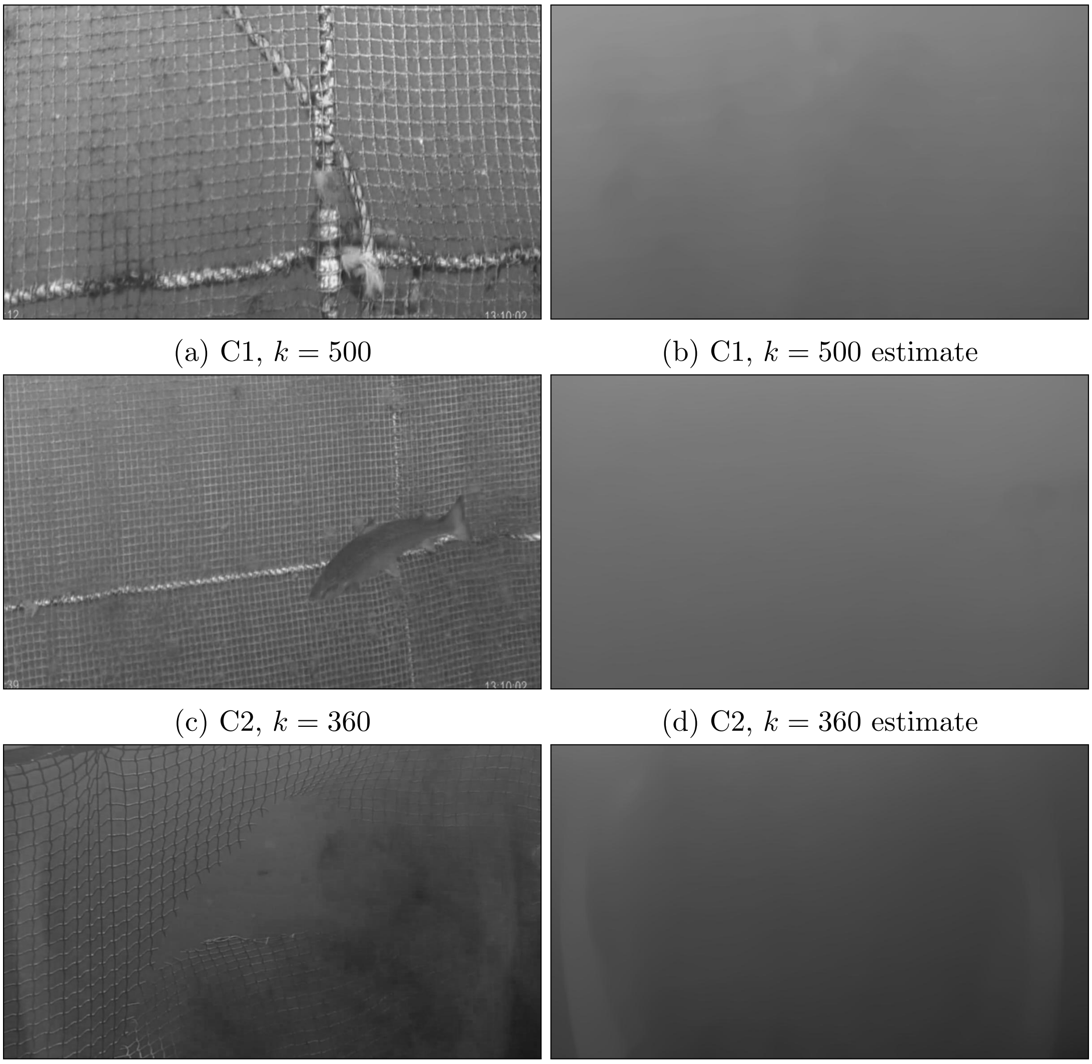
(d) C2, $k = 350$ estimate

(e) C2, $k = 300$

(f) C3, $k = 300$ estimate

Figure B.7: Blue background estimates from the TMSU model with $n = 10$, $m = 5$ and $w_b = 2$

B.2.3 TMCSU Background Estimates



(a) C1, $k = 500$

(b) C1, $k = 500$ estimate

(c) C2, $k = 360$

(d) C2, $k = 360$ estimate

(e) C3, $k = 300$

(f) C3, $k = 300$ estimate

Figure B.8: TMCSU background estimates from the blue color channel

B.3 Graphs

B.3.1 Error Correction Motion Graphs

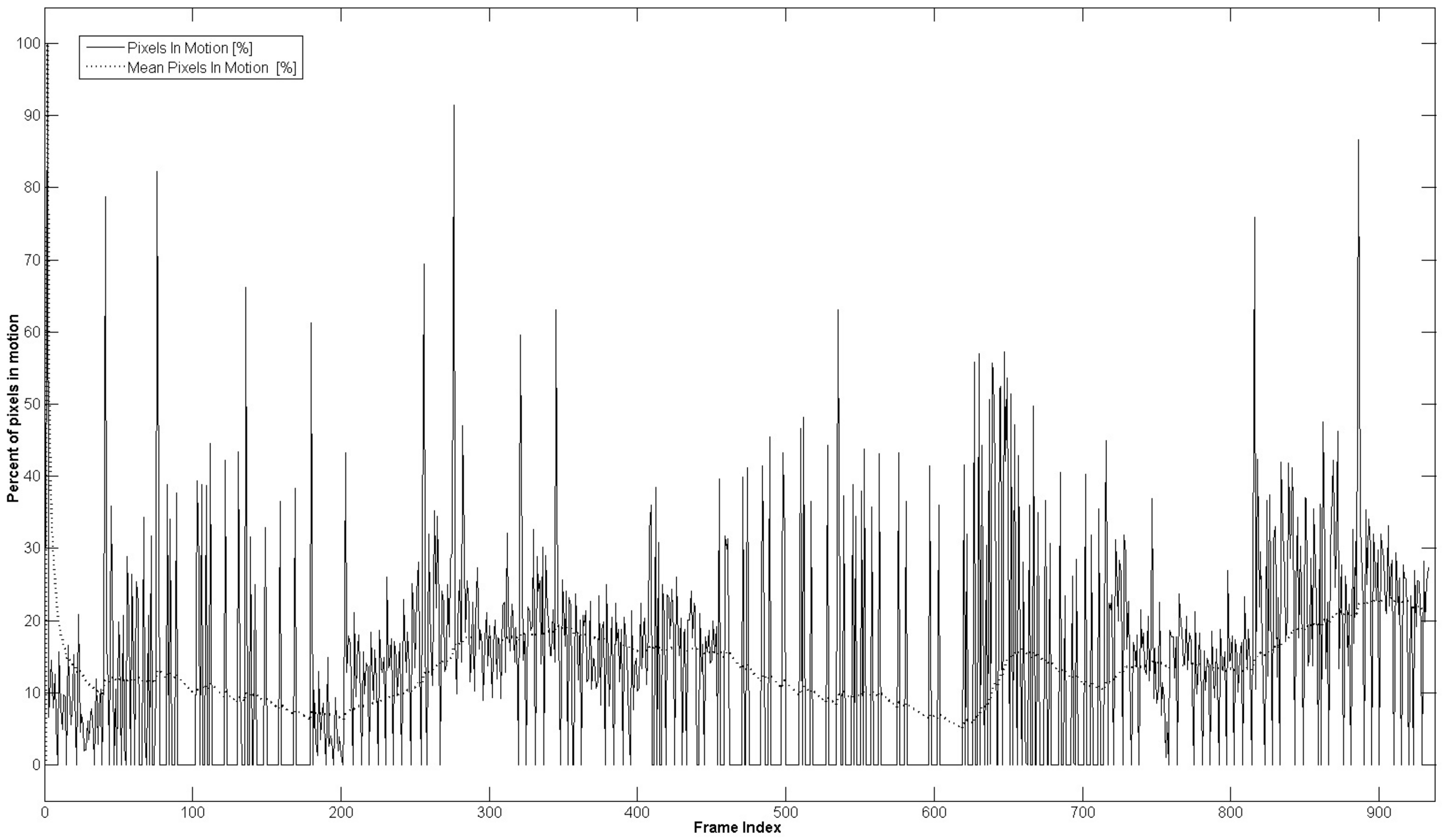


Figure B.9: Motion analysis of untreated video material C3

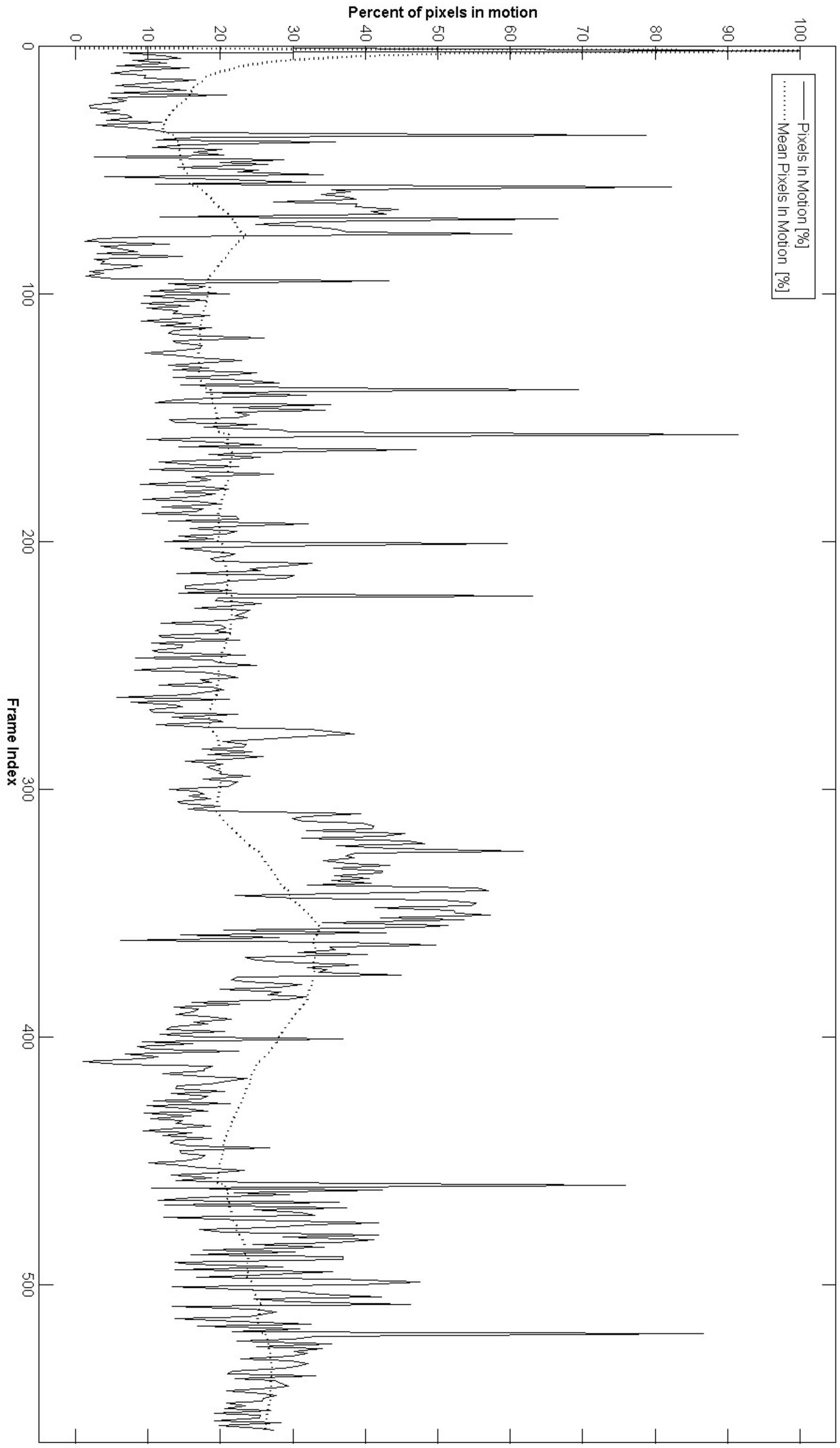


Figure B.10: Motion analysis of treated video material C3

B.3.2 TMSU Performance Graphs

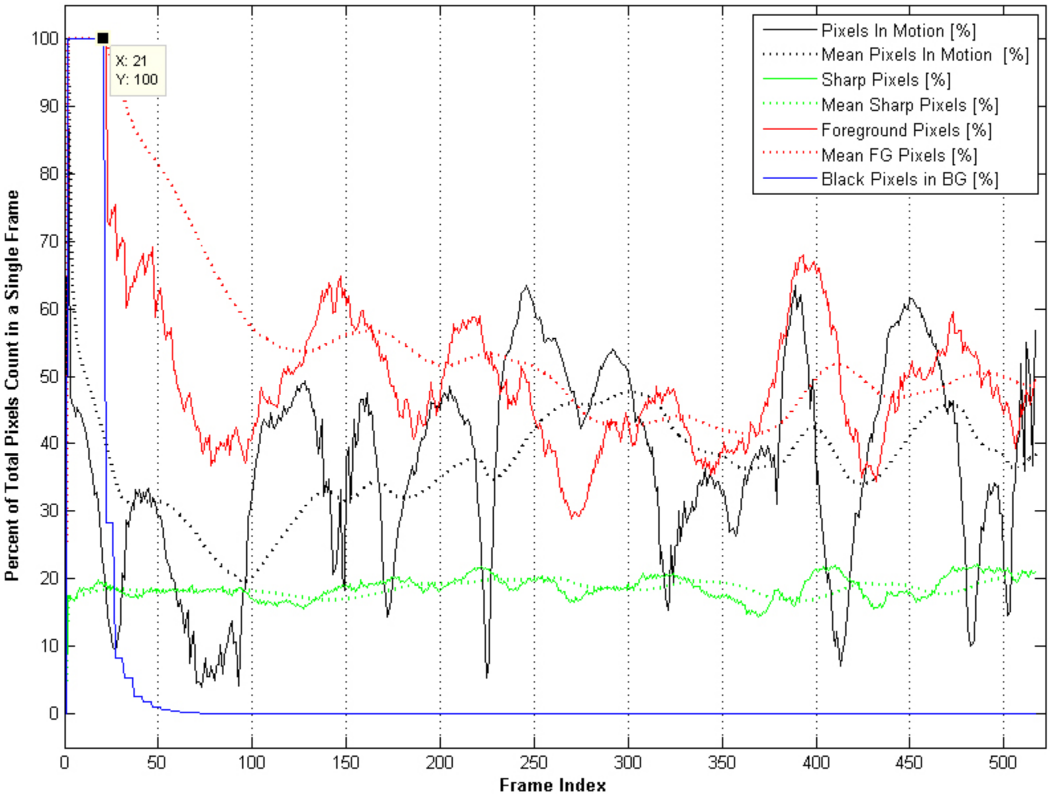


Figure B.11: TMSU graphs, C1 red channel, $n = 10$, $m = 5$, $w_b = 2$

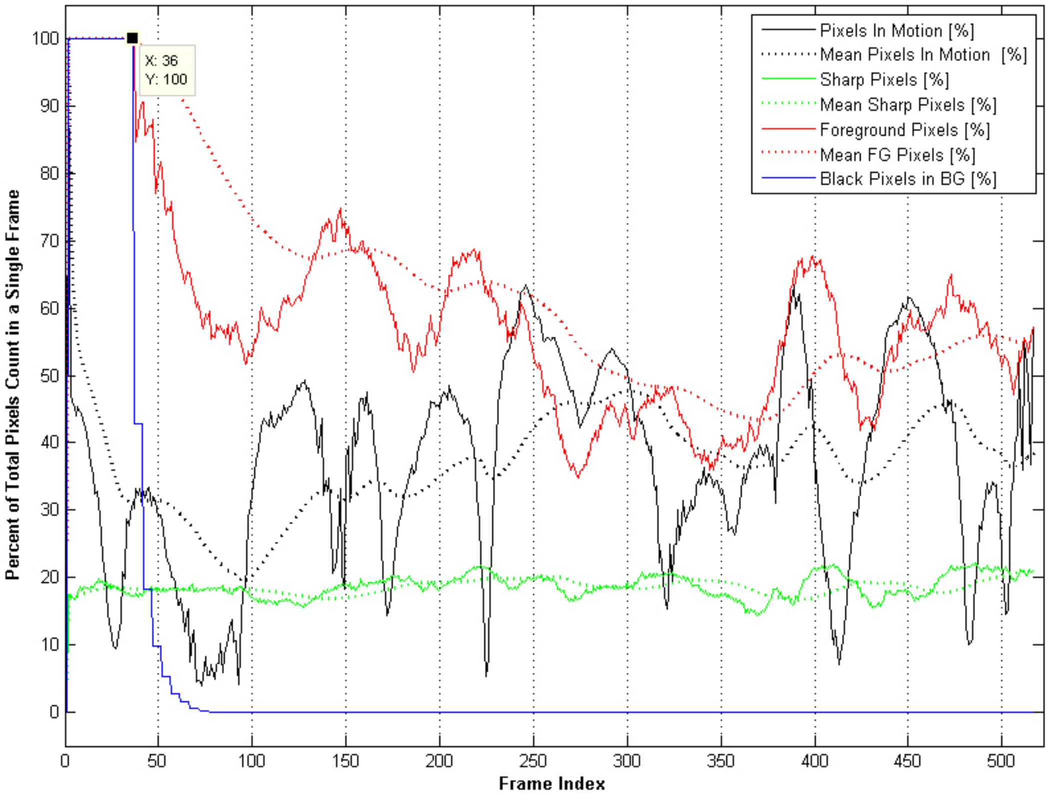


Figure B.12: TMSU graphs, C1 red channel, $n = 10$, $m = 5$, $w_b = 5$

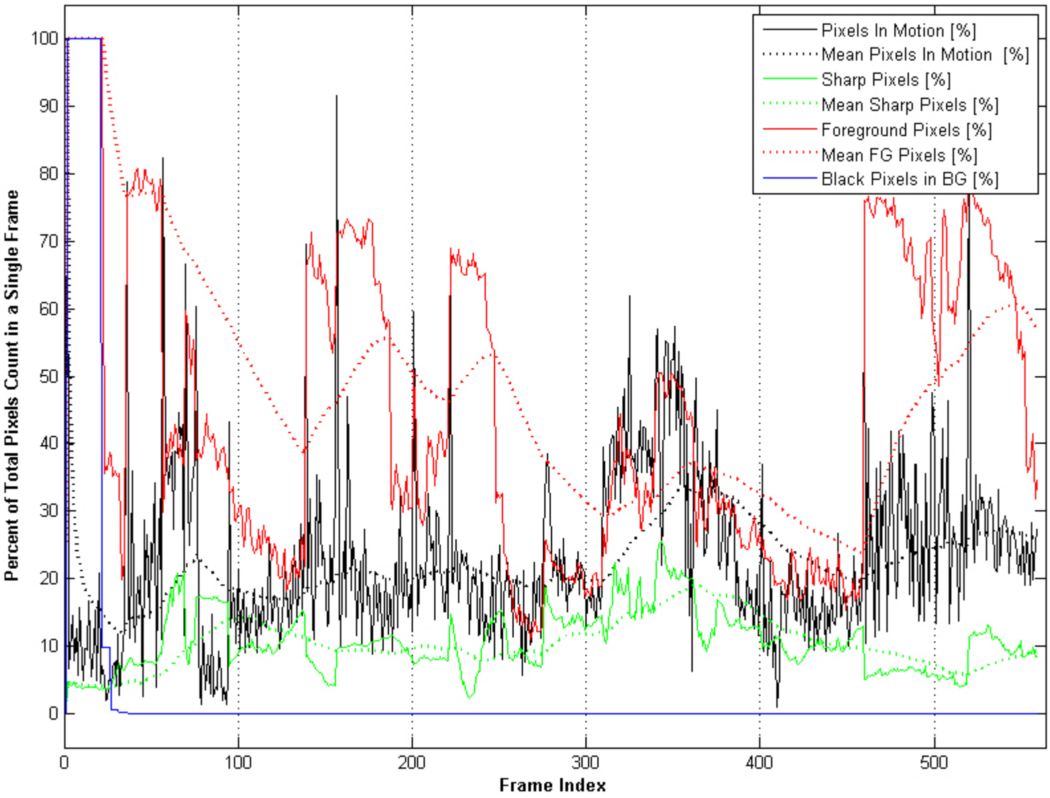


Figure B.13: TMSU graphs, C3 red channel, $n = 10$, $m = 5$, $w_b = 2$

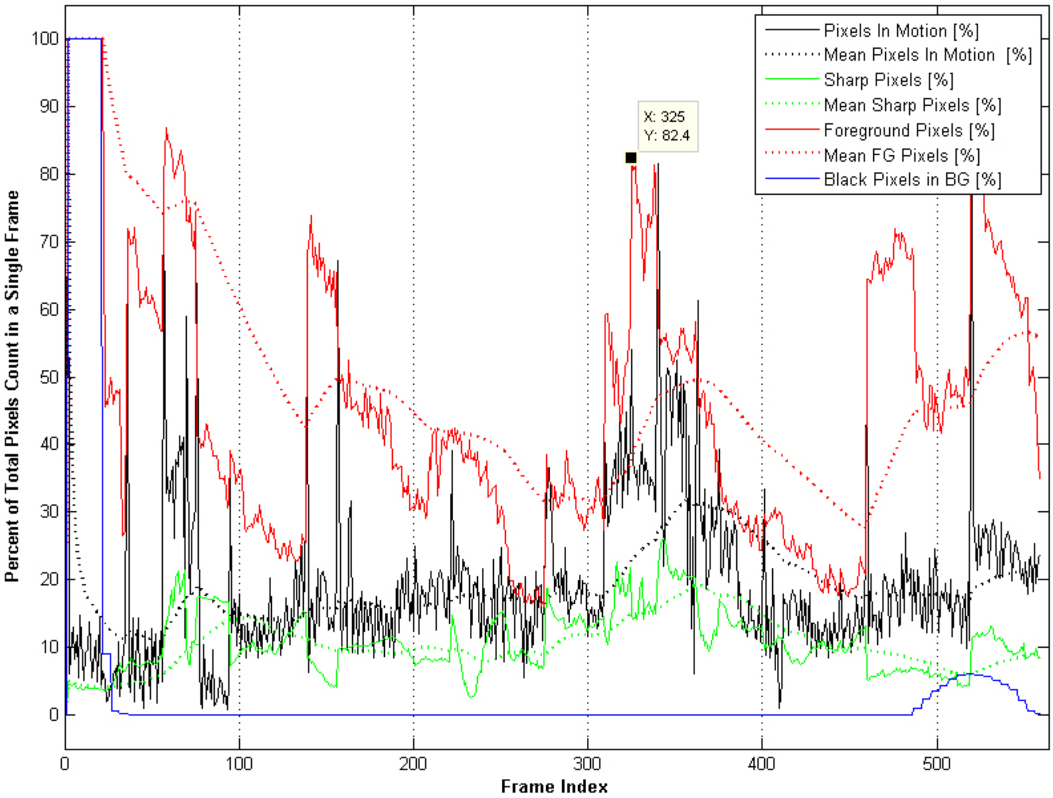


Figure B.14: TMSU graphs, C3 blue channel, $n = 10$, $m = 5$, $w_b = 2$

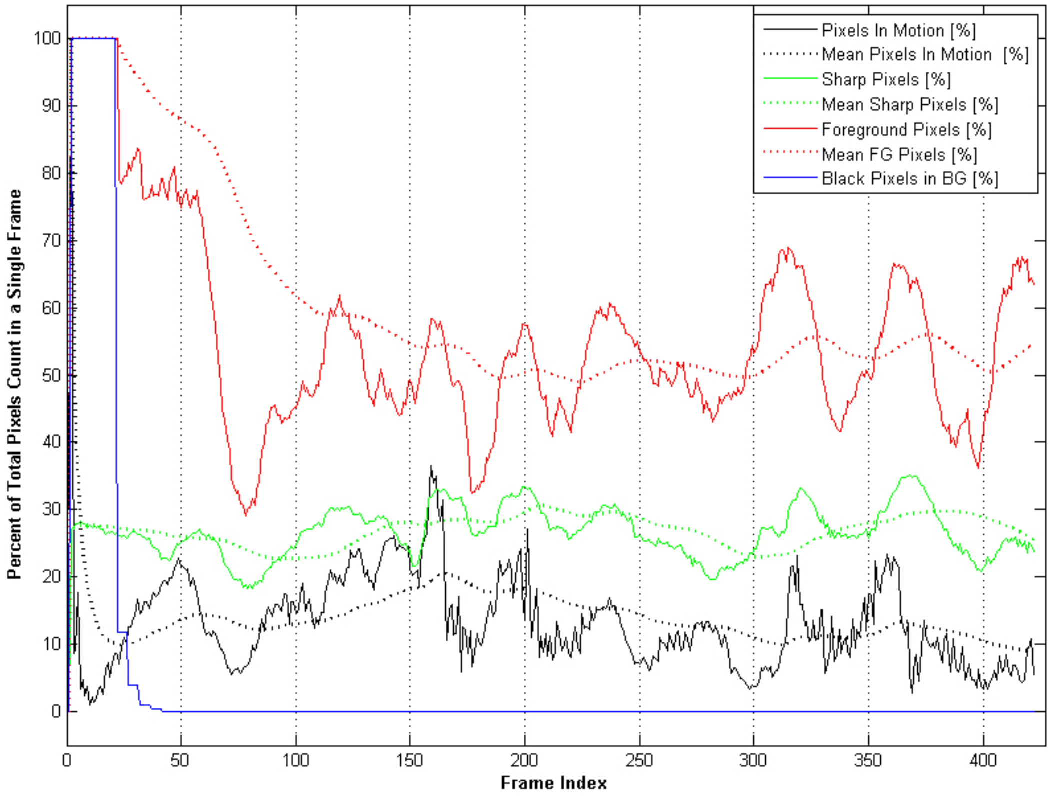


Figure B.15: TMSU graphs, C2 red channel, $n = 10$, $m = 5$, $w_b = 2$

B.3.3 Final System Performance Graphs

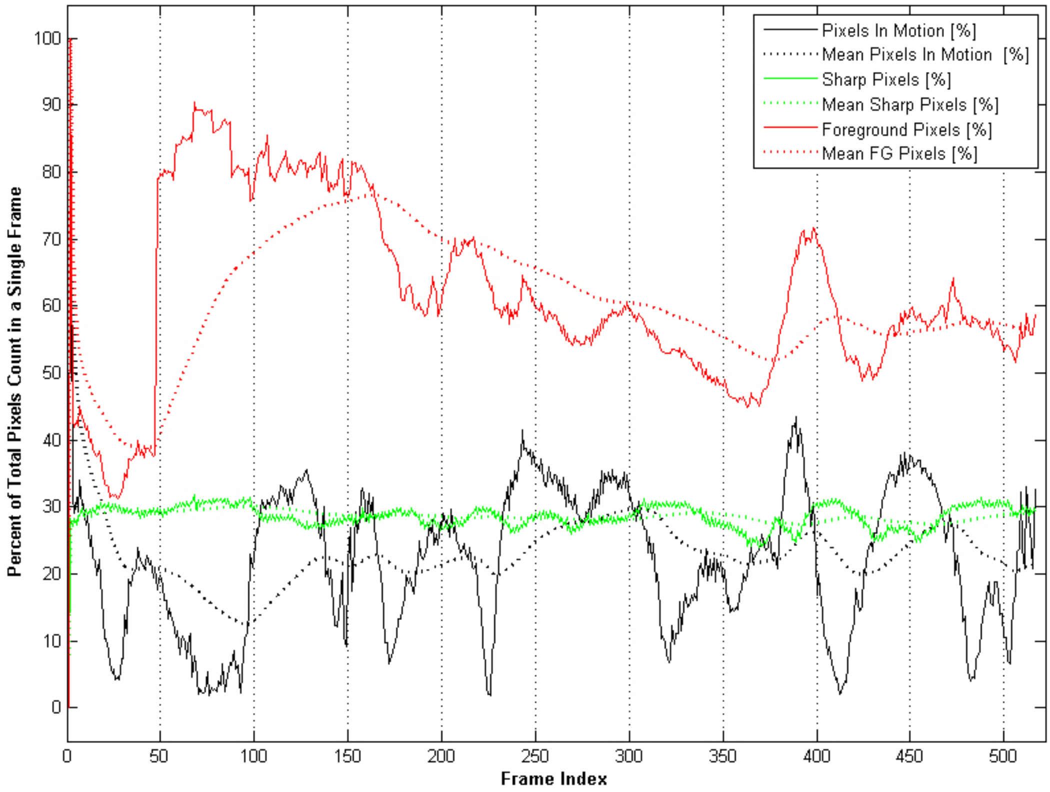


Figure B.16: Final system performance graph for C1

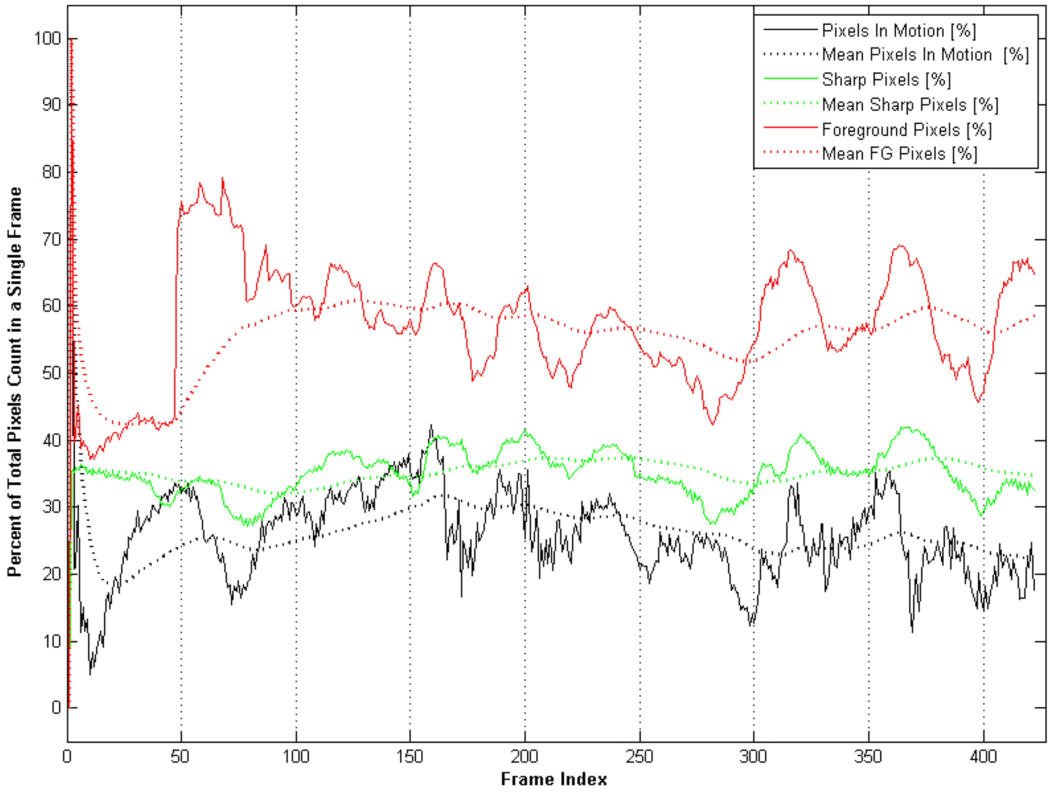


Figure B.17: Final system performance graph for C2

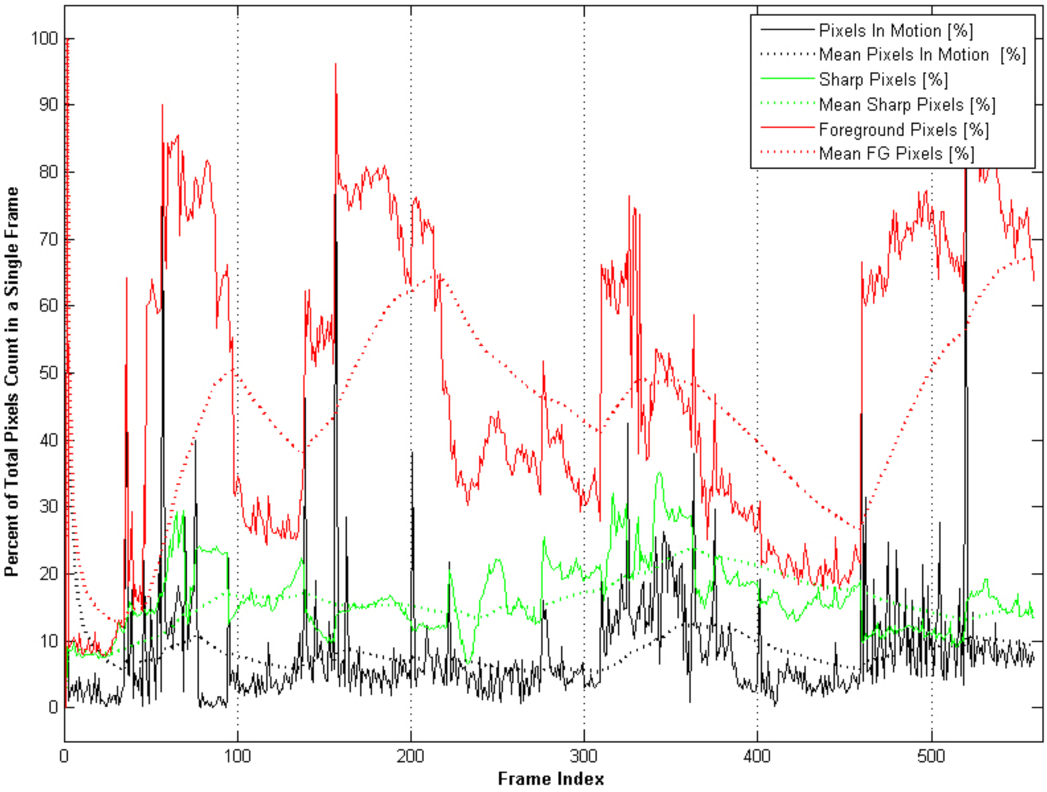


Figure B.18: Final system performance graph for C3

B.4 Parameter Tables

$n \rightarrow$ $w_b \downarrow$ (m)	**5**	**8**	**9**	**10**	**15**	**20**	**30**
0				(10)			
1	(10)						
2	(10)		(5,10)	(5,10, 20,30,60)	(5,10)	(10)	(10)
4		(10)		(10)		(20)	
5				(5,10)	(5,10)	(10)	
6				(10)			(30)
7				(5,10)	(5,10)		
8				(10)			
10	(60)						(10)

Table B.1: All parameter settings tested for the TMSU method