Lappeenranta University of Technology
School of Engineering Science
Intelligent Computing Major

Master's Thesis

**Ekaterina Lantsova**

# AUTOMATIC RECOGNITION OF FISH FROM VIDEO SEQUENCES

Examiners: Associate Professor Arto Kaarna
Associate Professor Tatiana Voitiuk

Supervisors: Associate Professor Arto Kaarna
Associate Professor Tatiana Voitiuk

# ABSTRACT

Lappeenranta University of Technology
School of Engineering Science
Intelligent Computing Major

Ekaterina Lantsova

**AUTOMATIC RECOGNITION OF FISH FROM VIDEO SEQUENCES**

Master's Thesis

2015

52 pages, 29 figures, 2 tables.

Examiners: Associate Professor Arto Kaarna
Associate Professor Tatiana Voitiuk

Keywords: machine vision, pattern recognition, automatic fish detection, object tracking.

The problem of automatic recognition of the fish from the video sequences is discussed in this Master's Thesis. This is a very urgent issue for many organizations engaged in fish farming in Finland and Russia because the process of automation control and counting of individual species is turning point in the industry. The difficulties and the specific features of the problem have been identified in order to find a solution and propose some recommendations for the components of the automated fish recognition system. Methods such as background subtraction, Kalman filtering and Viola-Jones method were implemented during this work for detection, tracking and estimation of fish parameters. Both the results of the experiments and the choice of the appropriate methods strongly depend on the quality and the type of a video which is used as an input data. Practical experiments have demonstrated that not all methods can produce good results for real data, whereas on synthetic data they operate satisfactorily.

# PREFACE

First of all, I would like to thank my Master's Thesis supervisor Arto Kaarna for his support and guidance during the difficult parts of this research work.

I am grateful to the whole organization "Kymijoen vesi ja ympäristö ry" for providing me with video materials, especially my contact person Janne Raunio.

I also thank the Kotka Maretarium staff for opportunity to make additional testing data for my research work.

Finally, I would like to thank Professor Tatiana Zudilova and Associate Professor Tatiana Voitiuk, Software Development Chair, Department of Infocommunication Technologies, ITMO University for support and encouragement in difficult times.

Lappeenranta, May 2015

*Ekaterina Lantsova*

# CONTENTS

# LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|---|---|
| BS | Background Subtraction |
| DCT | Discrete cosine transform |
| DoG | Difference-of-Gaussian |
| HMM | Hidden Markov Models |
| kNN | k-nearest-neighbor |
| MDS | Multi-Dimensional Scaling |
| SIFT | Scale Invariant Feature Transform |
| UAV | Unmanned aerial vehicle |

# 1 INTRODUCTION

## 1.1 Research problem

The fishing industry is highly developed in Finland and Russia. However, this area needs many processes to be automated, but the specific nature of the field causes certain difficulties in the performance of this task.

The aim of the thesis is to develop a solution of the adaptive pattern recognition, applicable in practice and capable of working with complex data types, such as three-dimensional scenes in video sequences. This development may be used to speed up operations such as identification and classification of the fish from video sequences. The testing video sequences examples are used as an initial data. In this research there are two types of video sequences for making experiments: the wildlife (real) video and synthetic video. The first type is a low quality video, which is made by a stable position camera in a special tube where the fish are moving, and the second type is made by professional camera and has a higher image quality. The methods which are described in this paper are implemented in both of these video types, but the performed experiments show different results.

## 1.2 Objectives and delimitations

The goal of this thesis is to study the opportunities of automated image recognition systems in relation to the recognition of the fish under water. Tasks such as object detection, object tracking and object recognition are considered. Effectively, the following questions are answered:

- "How the object if interest (in our case it is a fish) can be defined and detected from the video sequence? Which methods need to be implemented?"
- "How the object of interest can be tracked?"
- "Is it possible to estimate type of fish from the video sequences?"
- "Is it possible to estimate parameters (e.g. size) of the fish from the video sequences?"
- "What problems may occur in the testing phase while using the selected methods?"

The issues that are outside the scope of this work, such as the requirements for the technical components of the system or camera calibration problem are not considered.

## 1.3 Structure of the Thesis

The Master's thesis paper consists of six sections. Section one gives the introduction and the task statement. The related work survey can be found in section two. The most significant methods are reviewed in section three. Section four contains description of the initial data and preprocessing and implemented methods of the performed experiments. In the last two sections the results of experiments and future work are described and analysed.

# 2 BACKGROUND

## 2.1 Object recognition and object tracking tasks

The task of object recognition in the field of computer vision means the finding of a given object in an image or video sequence. This problem is the one of the most complicated targets for computer vision systems in the modern world. Suchlike tasks are trivial for Humans, because humans can recognize objects even if they are rotated, translated, scaled or partially obstructed from the view. There are many approaches which can be applied to object recognition in single images or still images of the object taken from different perspectives and in different poses. Some of the challenges posed by this particular task include recognizing the object when looking at it from a different perspective and pose, recognizing the object when it is partially occluded and tracking the object while it is in motion (Guo, 2001).

Videos are actually sequences of images (Figure 1), each of which is called a frame, displayed in fast enough frequency so that human vision system percepts the continuity of its content. It is obvious that all image processing techniques can be applied to individual frames. Besides, the contents of two consecutive frames are usually closely related.

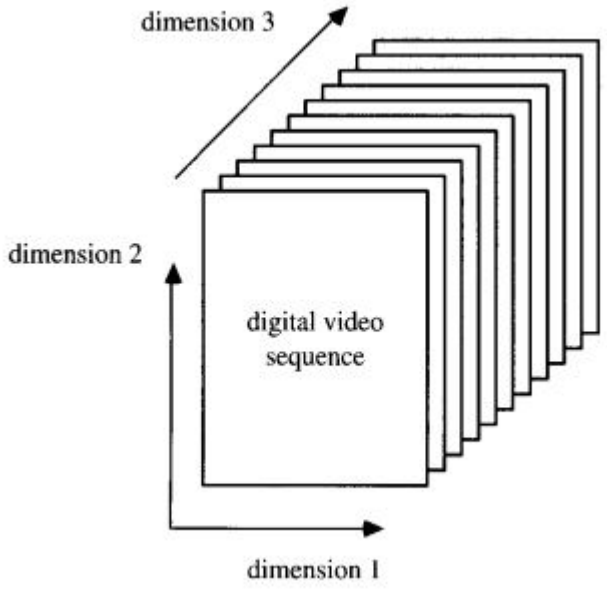



**Figure 1.** The dimensionality of images and video, where dimensions 1 and 2 are spatial dimensions and dimension 3 is a time (Bovik, 2005).

Visual content can be modeled as a hierarchy of abstractions. At the first level are the raw of pixels with color or brightness information. Further processing yields features such as edges, corners, lines, curves, and color regions. A higher abstraction layer may combine and interpret these features as objects and their attributes. At the highest level are the human level concepts involving one or more objects and relationships among them (Guo, 2001).

Following are the basic steps (Figure 2) for tacking an object, as described in literature (Parekh et al., 2014):

- Object detection

  Object detection is identifying objects of interest in the video sequence. Object detection can be done by various techniques such as frame differencing, optical flow and background subtraction.

- Object Classification

  Object can be classified as vehicles, birds, floating clouds, swaying tree and other moving objects. The approaches to classify the objects are e.g. shape-based classification, motion-based classification, color based classification and texture based classification.

- Object Tracking

  Tracking can be defined as the problem of approximating the path of an object in the image plane as it moves around a scene. The approaches to track the objects are point tracking, kernel tracking and silhouette.

Following challenges should be taken care in object tracking as described in (Athanesious et al., 2012):

- Loss of evidence caused by estimate of the 3D realm on a 2D image.
- Noise in an image.
- Difficult object motion.
- Imperfect and entire object occlusions.
- Complex objects structures.

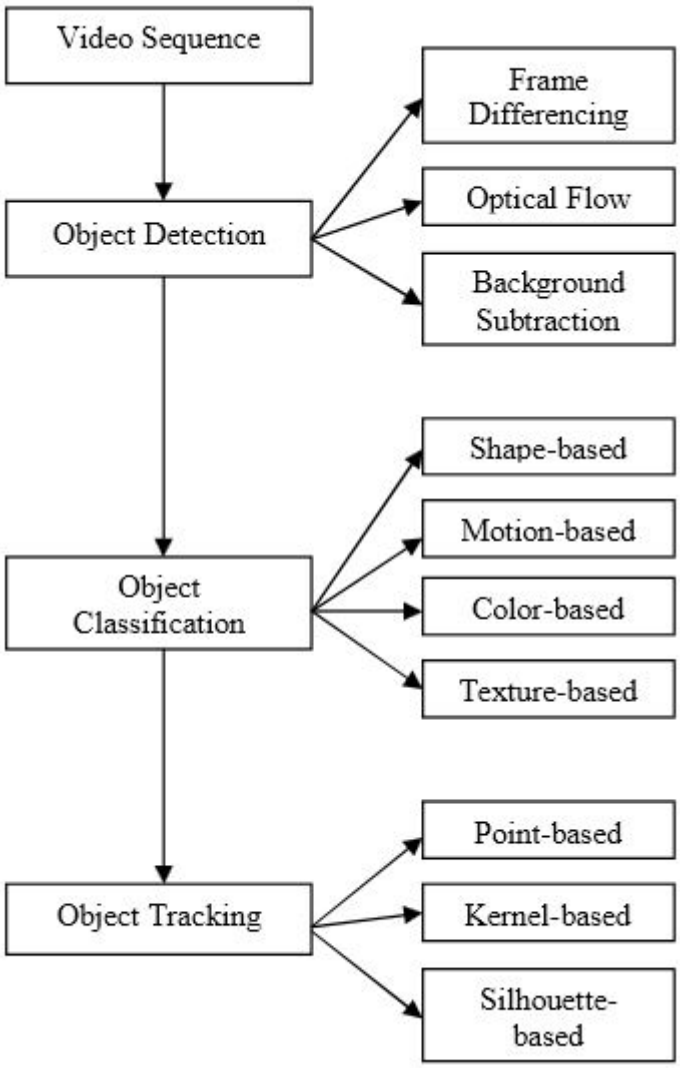



**Figure 2.** Basic steps for tracking an object (Lee et al., 2011).

## 2.2 Related work in automatic recognition of animals and fish

Nowadays the problem of recognition animals and fish from images or video sequences is a very common and complicated task. The biologists usually need to investigate a large amounts of video files during the studies of behavior of animals, birds, fish and plants. Such type of the task is a time-consuming because very often the data is not sufficiently indexed. Therefore, computer-based visual analysis methods are able to considerable accelerate the process of video indexing and searching in the large video collections (Zeppelzauer, 2013).

In the related work (Zeppelzauer, 2013) a fully automated method for the detection and tracking of elephants in wildlife video which has been collected by biologists in the field

was proposed. The main idea is to represent a solution for the problem of automated detection of elephants in wildlife by using color-based object detection methods and object tracking methods. The method which was used in (Zeppelzauer, 2013) dynamically learns a color model of elephants from a few training images. It localized elephants in video sequences with different backgrounds and lighting conditions based on the color model. The goal is to detect elephants (or groups of elephants) performing different activities. The animals can be of different sizes and poses. Performed experiments showed that both near- and far-distant elephants can be detected and tracked reliably. The method (Zeppelzauer, 2013) does not make hard constraints on the species of elephants themselves. Therefore, same method can be easily adaptable to other animal species.

Very often in real-life settings with unconstrained video material, the detection of animals by specialized detectors becomes unsuitable and does not work stable due to the large number of unpredictable environmental influences, such as occlusions, lighting variations, and background motion. For this reason only a limited number of approaches has been introduced that faces the challenges of unconstrained wildlife video (Zeppelzauer, 2013).

Figure 3 represents the basic steps of the approach (Zeppelzauer, 2013): "First, a color model is generated from labeled ground truth images. Next, image segments are classified by the color model. Positively detected segments (candidates) are tracked through the sequence resulting in spatiotemporally coherent candidates. The final detections are obtained by validating the spatiotemporal candidates by shape, texture, and consistency constraints. Finally, postprocessing fills gaps in tracking for each detection."

The elephants are detected with high accuracy, but at the same time many false-positive detections are generated (Zeppelzauer, 2013): "The mean color seems to be a suboptimal representation that removes too much information about the color distribution in the segments. To compensate for this limitation, we propose a more fine-grained two-stage classification that operates on the individual pixels of a segment."

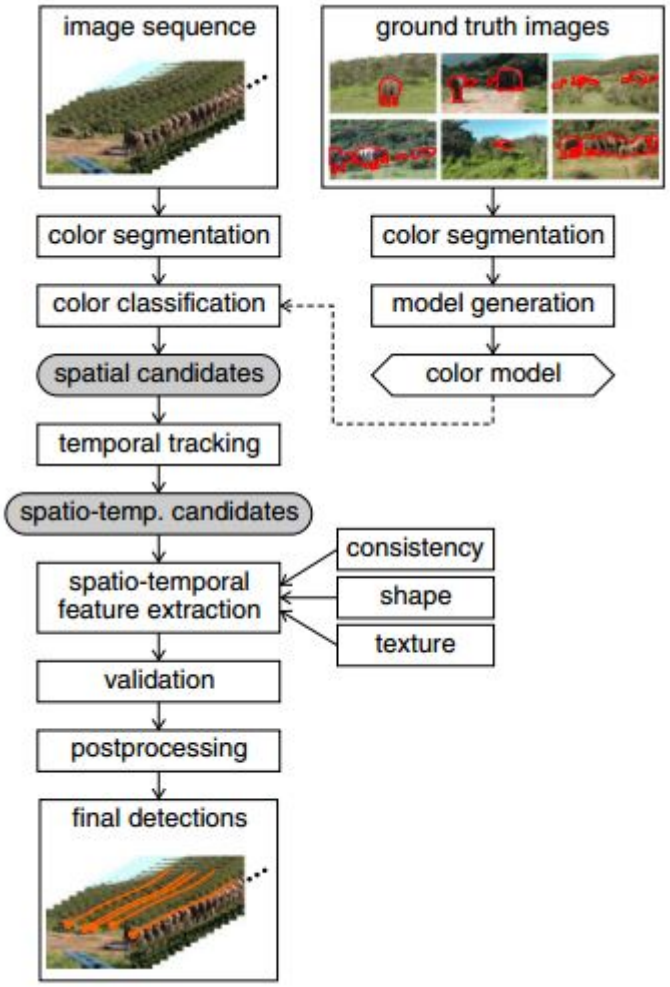



**Figure 3.** Overview of automated elephant detection (Zeppelzauer, 2013).

According to this idea, at first, each pixel was classified by the classifier used in one-stage classification and then voting to the individual predictions was applied. If the percentage of positively classified pixels was above two thirds, the segment was classified as positively detected; otherwise, the entire segment was reject. Results in Figures 4 and 5 show that the two-stage classification is more robust in false detections while it detects elephants equally well. In Figure 4 the input image represented in image (a), the results of one-stage classification presented in images (b,c), and results of two-stage classification in images (d,e). Positively detected regions are highlighted by red contours in (b) and (d). The remaining segments in the image are shown in (c) and (e).

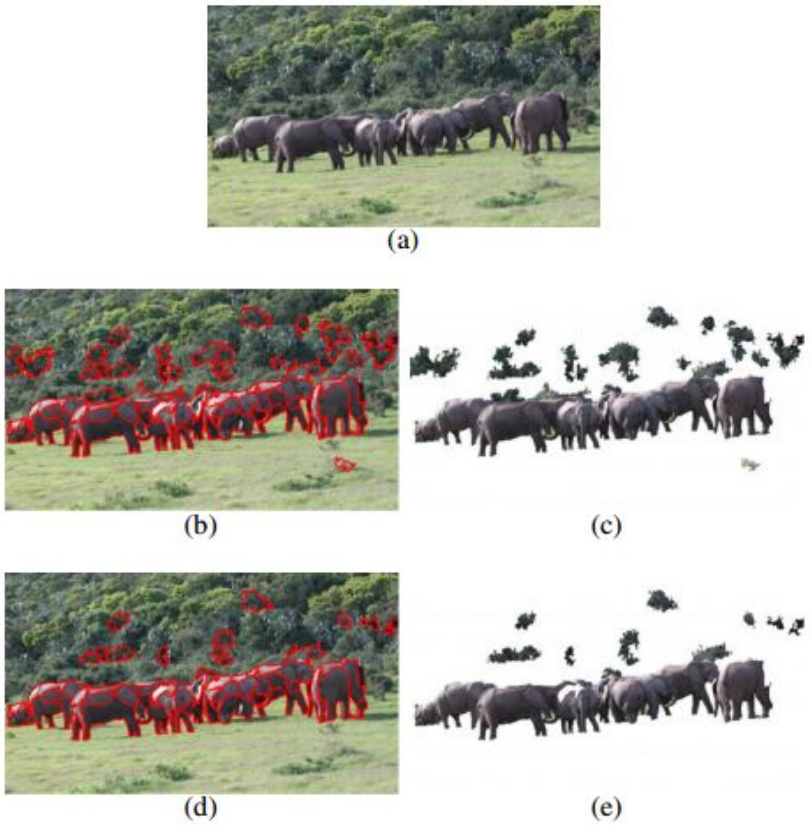


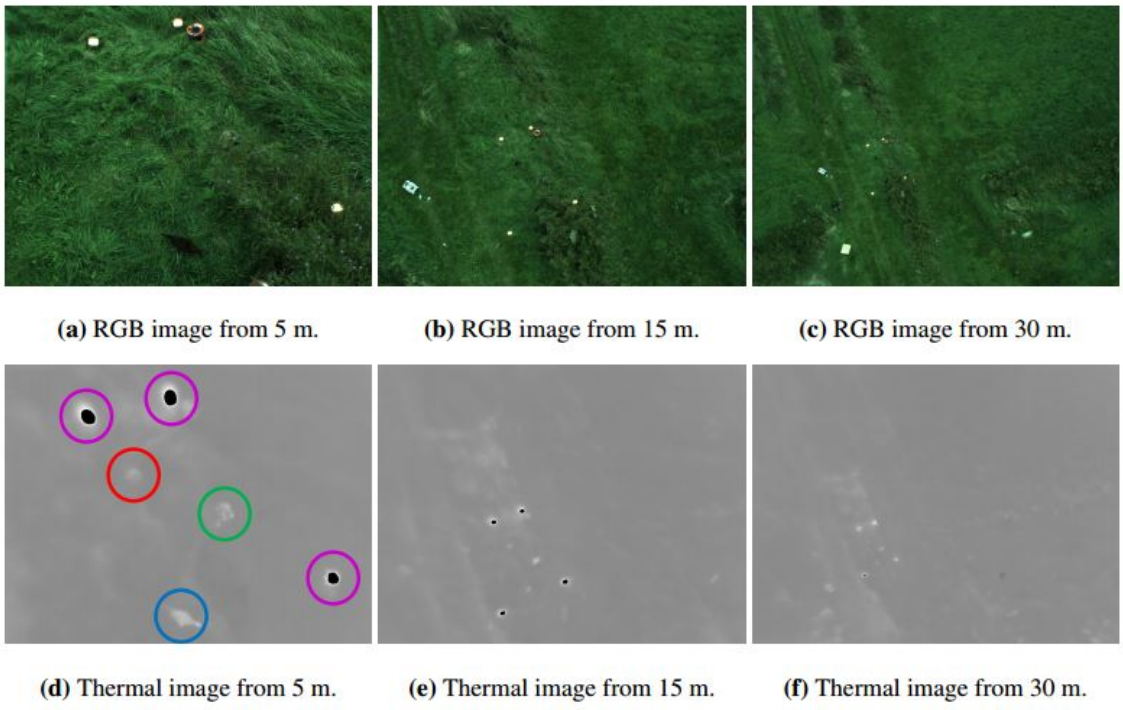

**Figure 4.** Color classification of an input image using two different classification schemes (Zeppelzauer, 2013).

**Figure 5.** Ground truth for different sequences. Left is original image, in the middle is labeled image, and right is ground truth the mask (Zeppelzauer, 2013).

Another related work is presented in (Christiansen et al., 2014) where the videos are captured using thermal camera. In (Christiansen et al., 2014) the problem of wildlife mortality in agricultural mowing operations were considered. Thousands of animals are injured or killed each year, due to the increased working widths and speeds of agricultural machinery.

Several approaches and methods are proposed to reduce this wildlife mortality: "The work presented in this paper contributes to the automated detection and classification of animals in thermal imaging. Hot objects are detected based on a threshold dynamically adjusted to each frame. For the classification of animals, we propose a novel thermal feature extraction algorithm. For each detected object, a thermal signature is calculated using morphological operations. The thermal signature describes heat characteristics of objects and is partly invariant to translation, rotation, scale and posture. The discrete cosine transform (DCT) is used to parameterize the thermal signature and, thereby, calculate a feature vector, which is used for subsequent classification." (Christiansen et al., 2014)

Figure 6 represents a visual RGB and thermal images. The same scene was captured from distance 5 m (a), 15 m (b) and 30 m (c). The scene consists of four halogen spotlights which can be easily visible in all images, a molehill, a rabbit and a chicken. A molehill, a rabbit, a chicken and three halogen spotlights are marked in image (d).



**(a)** RGB image from 5 m. **(b)** RGB image from 15 m. **(c)** RGB image from 30 m.

**(d)** Thermal image from 5 m. **(e)** Thermal image from 15 m. **(f)** Thermal image from 30 m.

**Figure 6.** Visual RGB and thermal images (Christiansen et al., 2014).

In (Christiansen et al., 2014) the methods for classification using measurements from both single and multiple frames are presented. The best performance can be achieves by combining measurements from multiple frames, with a balanced classification accuracy of 93.5% in the altitude range of 3-10 m and 77.7% in the altitude range of 10-20 m. The results demonstrate a clear relationship between the performance of detection and classification relative to altitude: "The simulated and limited dataset is favorable in terms of performance for the given algorithms. The actual applicability of the system should therefore be determined using footage from an actual UAV. The proposed detection and classification scheme is based on top-view images of wildlife, as seen by a UAV. The use of UAV-technology for automatic detection and recognition of wildlife is currently part of ongoing research towards wildlife-friendly agriculture."(Christiansen et al., 2014)

Another work proposes an automatic fish classification system that operates in the natural underwater environment (Spampinato, Giordano, et al., 2010). This system should assist marine biologists to understand fish behavior. The study considers the fish classification problem.

The two types of features perform a fish classification: the first is a texture features which were extracted by using statistical moments of the gray-level histogram, spatial Gabor filtering and properties of the co-occurrence matrix and second is a shape features extracted by using the curvature scale space transform and the histogram of Fourier descriptors of boundaries. An affine transformation is applied to the acquired images to represent fish in 3D by multiple views for the feature extraction (Spampinato, Giordano, et al., 2010).

Figure 6 shows the output of the detection system, where a bounding box is drawn around each fish. The first step of the proposed system aims at extracting trajectories by tracking fish over consecutive frames. The tracking system firstly automatically detects fish by means of a combination of the Gaussian mixture model and moving average algorithms, then tracks the fish by using the adaptive mean shift algorithm. The obtained accuracy for both fish detection and tracking is about 85% (Spampinato, Giordano, et al., 2010).

The results of system implementation are described as: "The system was tested on a database containing 360 images of ten different species achieving an average correct rate of about 92%. Then, fish trajectories, extracted using the proposed fish classification combined with a tracking system, are analyzed in order to understand anomalous behavior. In detail, the tracking layer computes fish trajectories, the classification layer associates trajectories to fish species and then by clustering these trajectories we are able to detect unusual fish behaviors to be further investigated by marine biologists."(Spampinato,

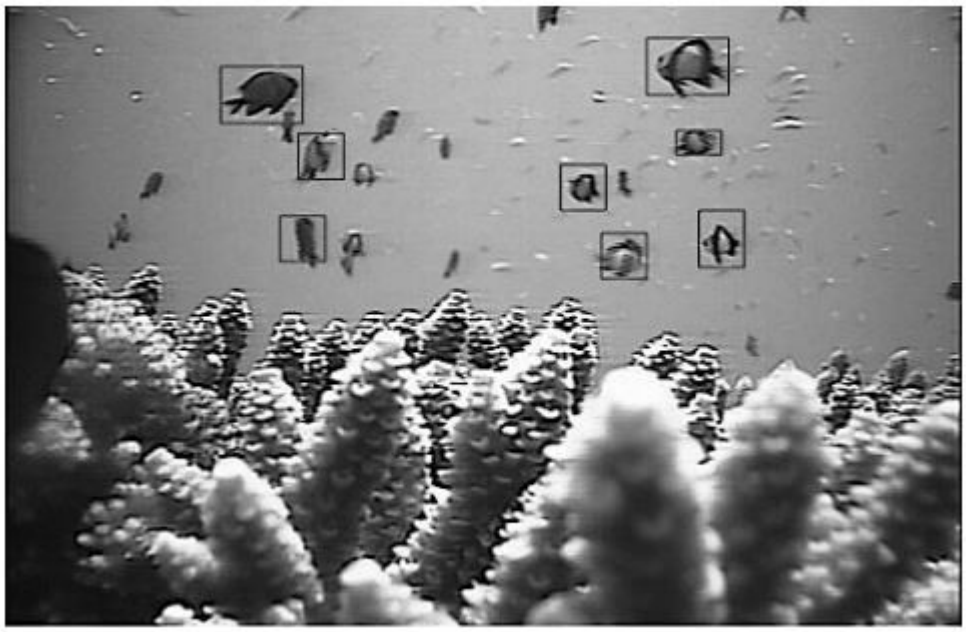


**Figure 7.** Output of the detection system (Spampinato, Giordano, et al., 2010).

Giordano, et al., 2010)

The approach which is used in the analysis of fish trajectories (Spampinato and Palazzo, 2012) based on the work which is proposed in (Suzuki et al., 2007). The characteristics of fish motion makes difficult estimation of events in terms of a sequence of simple moves. Such approach (Suzuki et al., 2007), which belongs to "clustering" category, makes it suitable to study the fish trajectories (Figure 8).

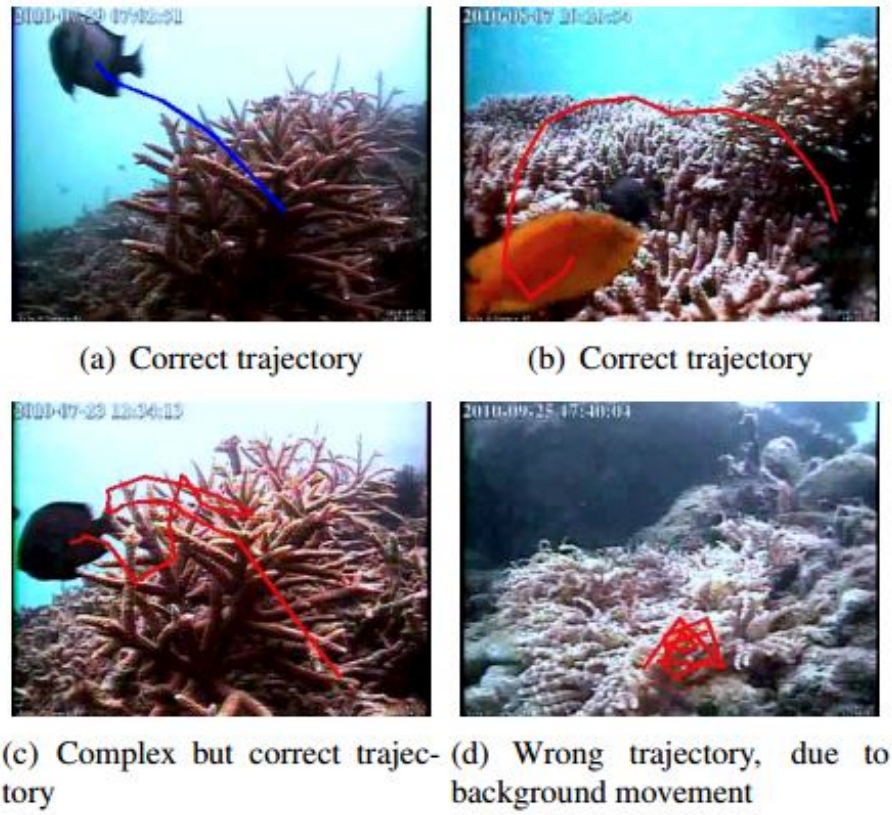



**Figure 8.** Examples of fish trajectories. (a) and (b): simple trajectories; (c): complex but still correct trajectory; (d): a wrong trajectory, due to background plant movements (Spampinato and Palazzo, 2012).

The basic idea of (Spampinato and Palazzo, 2012) is to use hidden Markov models (HMMs) to represent trajectories in a uniform way, without having to deal with different path sizes while keeping the underlying trajectory's dynamics. Based on (Spampinato and Palazzo, 2012): "A metric for HMMs is introduced in order to build a similarity matrix between all objects in the learning set which is used by a multi-dimensional scaling (MDS) algorithm to project trajectories onto a lower-dimensional space, where it is more feasible to perform trajectory clustering to identify classes corresponding to common patterns. In order to decide whether a new trajectory is anomalous, for each cluster a corresponding HMM is built and used to check whether the input trajectory matches it; trajectories that do not match significantly any of all the identified clusters are therefore detected as anomalous."

The methods of computer vision widely spread in the fish industry. This problem is particularly relevant for the food industry. The two following related works are dedicated to the methods for solving the problem of determining of the fish quality.

In the related work (Misimi et al., 2007) the computer vision method was used for color evaluating of the Atlantic salmon fillets. The review showed that the automation of the fish processing using computer vision and other robotic equipment to replace human inspectors will bring savings in labor costs about $ 1 per kilogram of fish produced. One of the processing operations of fish is grading line salmon. It is usually assumed that the color of salmon products is one of the most important parameter of quality in fish processing. In addition to saving labor costs, a system of computer vision-based automated processing of fish can improve the quality of the product (Arnarson et al., 1988).

As in the similar systems used in food industry, the computer vision system consist of the illumination setup, camera and a PC. When the image is captured it can be sent to the computer for further processing. A computer is used to implement an algorithm that allows the feature extraction, segmentation, classification, and quantification of images and objects of interest are contained in those images. Feature extraction consists of selecting distinct features, which can be used to recognize patterns in the different categories (Duda et al., 2000). The image is subdivided into its constituent regions of interest by using segmentation process. The goal of segmentation is to simplify or change the representation of an image into its constituent regions of interest that is more meaningful and easier to analyze (Gonzalez et al., 2003).

In the (Misimi et al., 2007) the fish from two different fish processing plants (Marine Harvest and Salmar AS, Hitra, Norway) was used. This two groups have different condition factor, which can be estimated by formulas: $K = 10^5 W/L^3$ where $W$ is the weight of the fish in grams (g), and $L$ is the length of fish in millimeters (mm).

The classification of the fish can be performed in following rule: $K = 1$ means that the fish is a very long and thin, in that reason it can be considered as poor fish (Misimi et al., 2007); $K = 1.4$ are considered to be good fish or well proportioned; $K = 1.6$ are fish in excellent condition. In both groups the color of the fillets was evaluated by commission according to the Norwegian Standard NS 9402 (1994) for measuring color of Atlantic salmon. The evaluation was performed visually in the daylight with using Roche color cards.

As described in (Misimi et al., 2007) color analysis and classification of fillets according to Roche cards by computer vision were performed in red, green, and blue (RGB) and CIE (Lab) color space. Figure 9 depicts the sequence of the classification algorithm.

The fillet could be isolated from background and be considered as a single region of interest for further analysis. For the purpose of the work (Misimi et al., 2007), the seg-

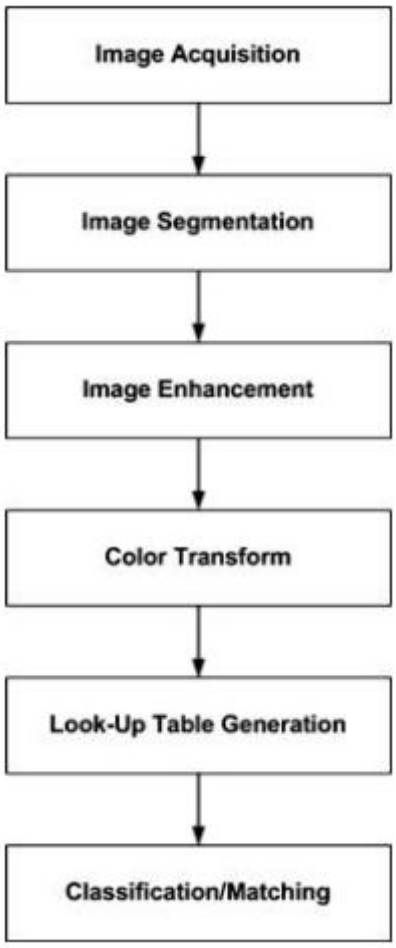



**Figure 9.** The scheme of the computer vision system for color evaluation (Misimi et al., 2007).

mentation was necessary for the purpose of color matching. The experiments showed that for both groups there are no significant differences in the color values according to Roche card were found between the computer vision method and the traditional method of sensory evaluation of color by human inspectors.

The main advantages of using computer vision to automate the sorting salmon are long-term working capacity and objectivity in the evaluation of color. This is possible because in the computer vision field, there is no eye fatigue or lack of color memory and lighting conditions are uniform. A computer vision system is able to process at least one fillet per second, whereas the human inspectors need a longer time. As was indicated above the introduction of such systems can save at 1$ per kilogram in labour costs.

One more related work refers to the fish industry. The article (Mathiassen et al., 2006) contains a description of a proof-of-concept prototype of an automated system for weight and quality grading of pelagic fish using a multi-modal machine vision system combined

with robotized sorting. Such type of the systems may be able to replace the majority of the manual operators needed today.

The reason for system implementation is an increased demand, and value attributed to higher quality of mackerel and herring. The demands usually include more precise weight class distribution and less damaged fish. The goal of the work (Mathiassen et al., 2006) is to find more accurate methods for weight and quality grading.

Nowadays weight grading is completed by using of V-belts or rotating rollers that separate the fish into four or more weight classes (Mathiassen et al., 2006). The accuracy of the such methods depends on the condition of the fish texture and the its body index. The catch quality also strongly depend on how the catching and handling has been carried through and from the skills of vessel crew. Today the quality grading today is done by the operators who control a fish species, check for over- or undersized fish and remove defective or damaged fish.

The suggested solution is designed for detecting defective whole pelagic fish such as herring and mackerel. The types of the defects are a superficial wounds indicated by broken fish skin exposing the underlying muscle and a scratches or scrapes without exposure of the underlying muscle.

Figure 10 depicts the scheme of the system prototype. The prototype consist of the conveyor belt (1), a robot (2), a diffuse illuminator (3), a laser (4) attached to the camera (5), a second laser (6) at an angle to the camera, and a computer workstation (7) (Mathiassen et al., 2006).

The machine vision system which is used for generating of the multimodal images consisting of gloss, scatter and 3D images. 3D images can be computed through triangulation of the image of laser line.

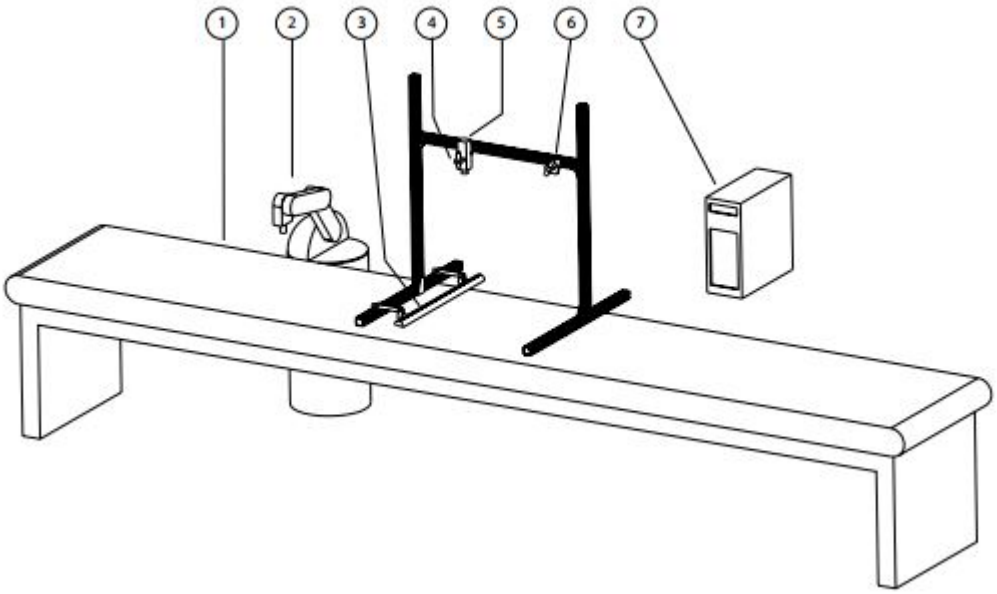



**Figure 10.** Overview of the prototype system for automatic sorting of pelagic fish (Mathiassen et al., 2006).

The 3d model is used to detect the correct orientation of the fish that makes optimum using of the information on the blaze and spread to detect surface defects. Also, this method is used to generate 3d images to detect defects such as the absence of head or tail. The 3d images were processed using computer vision algorithms to detect the fish defects.

The results of the developing of the prototype can be a sorting system which is able to detect of defects of the fish in real-time at a conveyor speed of 50 cm/s with a resolution in the multi-modal images of 0.3 mm in both dimensions of the conveyor plane and a height resolution of 0.25 mm in the 3D images. The conveyor speed limit is to 90 cm/s at this resolution (Mathiassen et al., 2006). Take in to account this requirement, the number of fish that can be checked for defects is limited only by the amount of fish that can be conveyed at 50 cm/s. The experiments showed a good quality high-resolution images of gloss and scatter that enhance the visibility of superficial wounds which were got from the multi-modal machine vision setup that enabled an automatic weighing of the fish. Using the present system, balance the classification will become more accurate than with the current use of the mechanical graders (Mathiassen et al., 2006).

# 3 METHODS

Moving object detection in a video is the process of identifying different object regions which are moving with respect to the background. More specifically, moving object detection in a video is the process of identifying those objects in the video whose movements will create a dynamic variation in the scene.

## 3.1 Background subtraction

Segmentation of moving regions in image sequences in real-time is a fundamental step in many automated visual surveillance systems. Background subtraction is one of the simple and typical methods which is used to segment moving regions in image sequences by comparing each new frame to a model of the scene background. Nevertheless, this method cannot distinguish between moving shadows and moving objects.

In (Kaewtrakulpong et al., 2001) a shadow detection scheme is introduced. As written in (Kaewtrakulpong et al., 2001) a background subtraction involves calculating a reference image, subtracting each new frame from this image and thresholding the result. Result is a binary segmentation of the image which highlights regions of non-stationary objects. A time-averaged background image is the simplest form of the reference image. The method which was considered in (Kaewtrakulpong et al., 2001) exposed to such problems such as e.g. impossibility to cope with gradual illumination changes in the scene. This approach requires a training period absent of foreground objects. The motion of background objects after the training period and motionless of foreground objects during the training period would be considered as permanent foreground objects. These problems lead to the requirement that any solution must constantly reestimate the background model.

The basic scheme of background subtraction assumes subtraction of the image from a reference image that models the background scene. Typically, the basic steps of the algorithm are as follows (Horprasert et al., 1999):

- Background modeling constructs a reference image representing the background.
- Threshold selection determines appropriate threshold values used in the subtraction operation to obtain a desired detection rate.
- Subtraction operation or pixel classification classes the type of a given pixel, i.e., the

pixel is the part of background (including ordinary background and shaded background), or it is a moving object.

Background subtraction is important part of the algorithm in many computer vision applications such as surveillance tracking and human poses estimation.

## 3.2 Motion Analysis and Object Tracking

From the viewpoint of reducing the amount of computation one of the promising approaches is a method in which a computationally complex operation (e.g. the detection of the object) is performed as infrequently as possible and some of the object detection and object tracking algorithms are used. In this case, when the detection object is performed less frequently, (e.g. only for the first frame of a video sequence) the position of each object in a subsequent frame is determined on the basis of information about the previous position of the object and information of the current frame. Typically, this approach leads to a significant decrease in total computational cost (Frantz et al., 2013).

The important part of computer vision systems is the tracking algorithm of the object from video sequence over time. The task of the tracker is to assess the trajectory of the object. The easiest way to track the position of an object from a sequence of frames is to use pattern matching. Search of the object position in the subsequent frame is performed in a sliding window and uses of a measure of similarity (e.g. the Euclidean distance). The most efficient algorithms that belong to the family of tracking algorithms are based on the using of kernel. In the other words it uses an iterative procedure that maximizes some measure of similarity. As the analysis of the literature (Frantz et al., 2013, Maggio et al., 2011), the existing methods of tracking objects have a number of drawbacks. One of the major drawbacks is the low accuracy of determining the position of the object. The input video sequence and the tracker receives an initial position of the object or objects that can be evaluated automatically or specified by the user. The tracker can automatically obtain an estimate of position of an object on all subsequent frames.

Tracking methods can be divided into two classes: the methods which track only one possible path and the methods which track multiple paths, and the better one should be chosen. Unfortunately, the problem of tracking the position of an object can be difficult for the following reasons (Hartley et al., 2004):

- Non-stationary background: not only foreground objects can be moved, but also

elements of the background.

- Slow motion of foreground objects. Therefore, if the foreground object is stationary or moves slowly, it cannot be distinguished from the background.
- Variability of lighting conditions: stage lighting changes frequently depending on the time of day and weather.
- Movement of objects in the scene cannot be predicted in advance. Furthermore, the position of the camera relative to the scene may change from frame to frame.

In this way, a particular approach to tracking the object in the video sequence must make a number of assumptions:

- The position of the object from frame to frame changes slightly.
- The camera may be moved, but only parallel to the plane of the sensor.
- Moving objects vary in size, but not very much.

## 3.3 Kalman filter

The object, which has already been detected, next can be traced along its path. One of the methods, which can be used for object tracking is the Kalman filter. Kalman filter is an efficient recursive filter which is able to estimate the state vector of the dynamic system using a series of incomplete and noisy measurements. It was performed by R.E. Kalman In 1960 when he published his famous paper which describes a recursive solution to the discrete-data linear filtering problem(Kalman, 1960).

The algorithm works in two steps. In prediction step Kalman filter extrapolates values of the state variables and their uncertainties. In the second stage, according to measurements, obtained with an error, the result of extrapolation is clarified. Due to incremental nature of the algorithm, it can monitor the status of the object in real-time (without looking forward using only current measurements and information on the previous state and its uncertainty (Grewal et al., 2001).

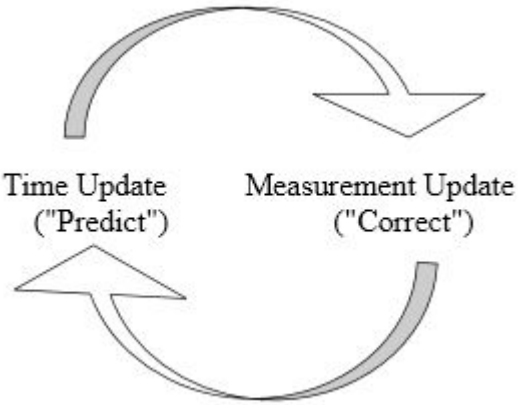



**Figure 11.** The Kalman filter discrete cycle (Welch et al., 1995).

The Kalman filter is an efficient recursive filter that estimates the internal state of a linear dynamic system from a series of noisy measurements (Faragher, 2012). Its model assumes the true state at time $k$ is evolved from the state at $(k-1)$ according to

$$\mathbf{x}_k = \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k \tag{1}$$

where

- $\mathbf{F}_k$is the state transition model which is applied to the previous state $\mathbf{x}_{k-1}$;
- $\mathbf{B}_k$is the control-input model which is applied to the control vector $\mathbf{u}_k$;
- $\mathbf{w}_k$is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance $\mathbf{Q}_k$ (Faragher, 2012).

$$\mathbf{w}_k \sim N(0, \mathbf{Q}_k) \tag{2}$$

At time $k$ an observation (or measurement) $\mathbf{z}_k$ of the true state $\mathbf{x}_k$ is made according to

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{3}$$

where $\mathbf{H}_k$ is the observation model which maps the true state space into the observed space and $\mathbf{v}_k$ is the observation noise which is assumed to be zero mean Gaussian white

noise with covariance $\mathbf{R}_k$.

$$\mathbf{v}_k \sim N(0, \mathbf{R}_k) \tag{4}$$

The initial state, and the noise vectors at each step $\{\mathbf{x}_0, \mathbf{w}_1, \ldots, \mathbf{w}_k, \mathbf{v}_1 \ldots \mathbf{v}_k\}$ are all assumed to be mutually independent.

The Kalman filter is a recursive estimator. This means that only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state. In contrast to batch estimation techniques, no history of observations and/or estimates is required. In what follows, the notation $\hat{\mathbf{x}}_{n|m}$ represents the estimate of $\mathbf{x}$ at time $n$ given observations up to, and including at time $m \leq n$.

The state of the filter is represented by two variables:

- $\hat{\mathbf{x}}_{k|k}$, the a posteriori state estimate at time $k$ given observations up to and including at time $k$;
- $\mathbf{P}_{k|k}$, the a posteriori error covariance matrix (a measure of the estimated accuracy of the state estimate).

The Kalman filter algorithm involves two stages: prediction and measurement update. The predict phase uses the state estimate from the previous timestep to produce an estimate of the state at the current timestep. This predicted state estimate is also known as the a priori state estimate because, although it is an estimate of the state at the current timestep, it does not include observation information from the current timestep. In the update phase the current a priori prediction is combined with current observation information to refine the state estimate. This improved estimate is termed the a posteriori state estimate.

The standard Kalman filter equations for the prediction stage are:

- Predicted (a priori) state estimate

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \tag{5}$$

- Predicted (a priori) estimate covariance

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k \tag{6}$$

The standard Kalman filter equations for the update stage are:

- Innovation or measurement residual

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \tag{7}$$

- Innovation (or residual) covariance

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{8}$$

- Optimal Kalman gain

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{9}$$

- Updated (a posteriori) state estimate

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \tag{10}$$

- Updated (a posteriori) estimate covariance

$$\mathbf{P}_{k|k} = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \tag{11}$$

The Kalman filter is optimal when the model perfectly matches the real system or the entering noise is white and the covariances of the noise are known.

## 3.4 Haar-like features for recognition

In 2001, P. Viola and M. John proposed the algorithm for adaptive face recognition, which became an innovation in the face recognition field. This method uses a sliding window. The frame of smaller size than the original image is moved over the input image with some step and a cascade of weak classifiers determines is there a face in this window or not. This sliding window method is effectively used in various problems of computer vision and object recognition (Viola et al., 2001).

One of the advantages of this methods is the possibility to recognize more than one face in the image. Although the method was developed for face recognition problem it can be also used for the recognition of the other objects. Also the using of simple classifiers shows a good speed of work which allows to find objects in real-time.

The method consist of two parts: algorithm of training and algorithm of recognition. In practice the rapidity of learning is not important. Nevertheless the learning algorithm requires a large amount of the test data and training can take days.

There are several principles which the method of Viola-Jones is based:

- Image used in the integral representation that allows to calculate quickly the required objects.
- Haar features are used by which the desired object is searched (in this context, and its face features).
- Boosting is used to select the most suitable characteristics for the desired object in this part of the image.
- All features are input to the classifier, which gives the result of "true" or "false".
- Cascades of features are used for rapid excluding of windows where the object of interest is not found.

### 3.4.1 Haar Cascades

Haar-like features are the signs of digital images used in pattern recognition. This name was intuitively chosen because of the similarity to Haar wavelets. Haar-like features were used in the first people faces detector working in real time. Historically, the algorithm that worked only with the intensity of the image (for example, RGB value in each pixel) has greater computational complexity. In the paper (Viola et al., 2004) the work with a variety of features, based on the Haar wavelet was considered. Viola and Jones adapted the idea of using Haar wavelets and developed what was called Haar-like features. Haar-like features consist of adjacent rectangular regions, as visualized in Figure 12. They are positioned in the image, then pixel intensities are summed in regions, and finally the difference between the sums is computed. This difference is the value of a particular trait, a certain size, a certain way to position the image.

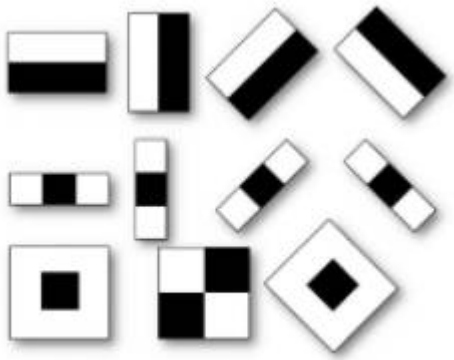


**Figure 12.** Haar-like features (Viola et al., 2004).

The features are represented by a set of rectangles (Figure 12). A rectangular Haar-like feature can be determined as the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the original image

The integral image $I(x, y)$ in location $x, y$ represents a sum of brightness values of the points for which the horizontal and vertical coordinates are smaller than $x$ and $y$. In the other words, each element of the integral image contains the sum of all pixels located on the up-left region of the original image in relation to the element's position (Simard et al., 1999):

$$II(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} I(x', y'), \tag{12}$$

where $II(x, y)$ is the integral image and $I(x, y)$ is the original image (Viola et al., 2004).

The integral image can be easily computed over the original image by following recurrences:

$$S(x, y) = S(x, y - 1) + I(x, y) \tag{13}$$

$$II(x, y) = II(x - 1, y) + S(x, y) \tag{14}$$

where $S(x, y)$ is the cumulative row sum, $S(x, -1) = 0$ and $II(-1, y) = 0$.

This allows to compute sum of rectangular areas in the image, at any position or scale, using only three integer operations:

$$\sum_{\substack{x0<x\leq x1\\ y0<y\leq y1}} i(x,y) = I(D) + I(A) - I(B) - I(C), \tag{15}$$

where points A, B, C, D belong to the integral image $I$, as visualized in the Figure 13 (Viola et al., 2004).

The algorithm uses a database of features for the object detection. It is possible to generate all combinations of Haar-Like features by screening of "weak" classifiers that give the error of the "second kind". Such mistake means that the object in the image is not find. One way to obtain the database is the AdaBoost algorithm which is described in (Viola et al., 2004).

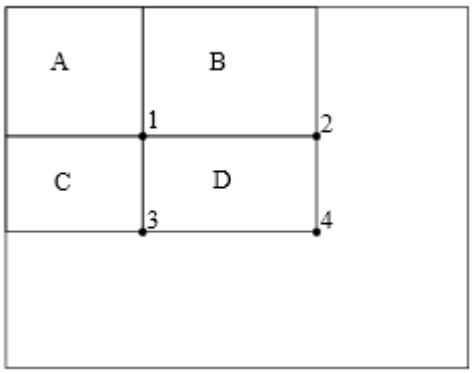



**Figure 13.** Integral representation of the image (Viola et al., 2004).

Cascade features consist of several stages. Each stage includes a set of features, which are divided into monochromatic rectangles, each of which assigned positive or negative weight. During the execution of the algorithm, a "window" size $W_h \cdot W_w$ pixels moves across the image horizontally and vertically. The initial size of the window is equal to the size of the window, recorded in the cascade classifier. At each step, the window size is increased, there are two ways to increase the window size. The first is to calculate the scaling factor and adjusting a rectangle inside features. The second is the scaling of the original image.

The disadvantages of the method are:

- Instability in changing lighting (possible solution is lighting neutralized normalization or transition to binarization area).

- Instability when zooming or rotating an image.
- Instability when the part of an image is a shifting background.

A key characteristic of the Haar-like features is a high speed compared with other methods. When the integral representation of the image is used a Haar-like features can be computed in constant time that allows to use classifiers in real time (Viola et al., 2001).

## 3.5 Scale Invariant Feature Transform (SIFT)

Image matching is a fundamental aspect of many problems in the field of computer vision. Large numbers of features can be extracted from typical images with efficient algorithms. The cost of extracting these features is minimized by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass an initial test. Among the variety of feature detection and description algorithms the Scale Invariant Feature Transform (SIFT) algorithm can be allocated as a method, which ensures that the features are invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine or 3D projection (Lowe, 2004).

Generally the SIFT algorithm consists of four steps (Lowe, 2004): scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptor.

- Detection of scale-space extrema

The cascade filtering approach is used for keypoints detection. The first stage of keypoint detection is identification of locations and scales that can be repeatably assigned under differing views of the same object (Lowe, 2004). In fact that it is impossible to use the same window to detect keypoints with different scale for larger corners the larger windows are needed. In SIFT, a scale-space filtering is used. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different scales (Figure 14). The scale space of an image can be defined as a function, $L\left(x, y, \sigma\right)$, that is produced from the convolution of a variable-scale Gaussian, $G\left(x, y, \sigma\right)$, with an input image, $I\left(x, y\right)$:

$$L\left(x, y,\sigma\right) = G\left(x, y,\sigma\right) * I\left(x, y\right), \tag{16}$$

where $*$ is the convolution operation in $x$ and $y$, and

$$G\left(x, y,\sigma\right) = \frac{1}{2\pi\sigma^2} e^{-\left(x^2+y^2\right)/2\sigma^2}. \tag{17}$$

Scale-space extrema in the difference-of-Gaussian function convolved with the image:

$$D\left(x, y,\sigma\right) = \left(G\left(x, y,k\sigma\right) - G\left(x, y,\sigma\right)\right) * I\left(x, y\right) \tag{18}$$

$$= L\left(x, y,k\sigma\right) - L\left(x, y,\sigma\right) \tag{19}$$

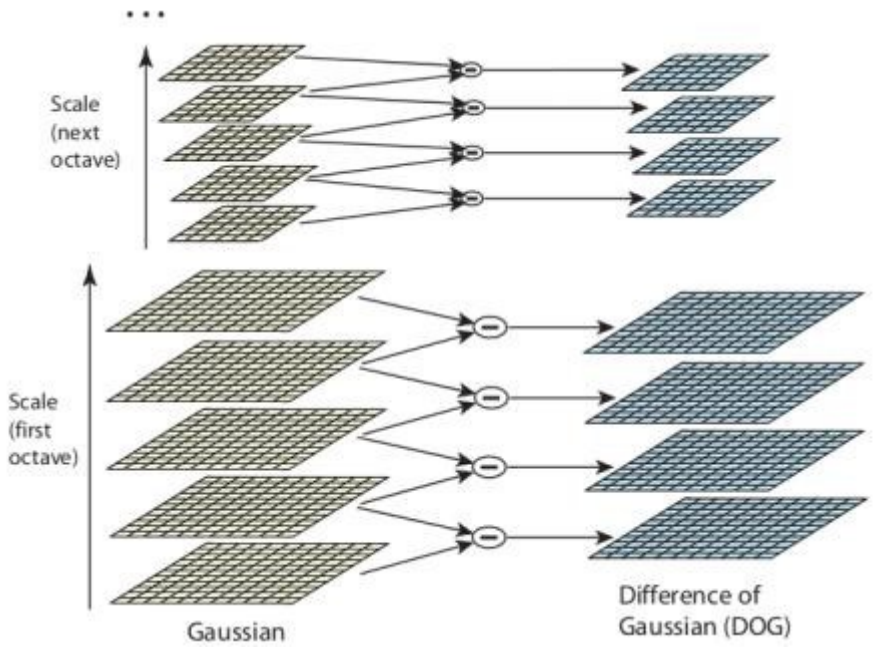



**Figure 14.** Scale space filtering (Lowe, 2004).

The candidates for the keypoints are the local extremes which were found in the DoG between two neighboring scales. For the local maxima and minima of $D\left(x, y, \sigma\right)$ detection, each pixel is compared to its eight neighbors in the current image and nine neighbors in the scale above and below (Figure 15). If it is large or smaller than all of this neighbors it can be selected (Lowe, 2004).

- Key point localization

When the potential keypoints locations are found, they have to be refined to get more accurate results. Taylor series expansion of scale space is used to get more accurate

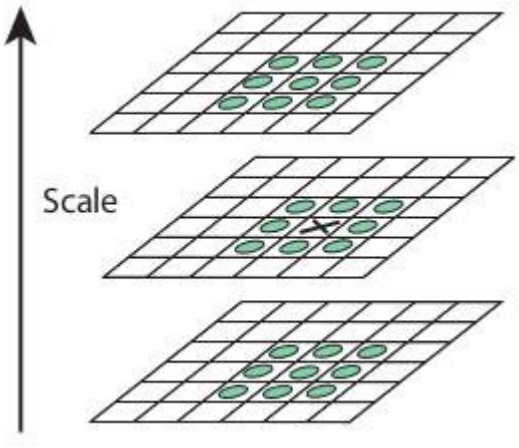



**Figure 15.** Potential keypoint (Lowe, 2004).

location of extrema. If the intensity at the extrema is less than a threshold value, it is rejected. DoG has higher response for edges, in this reason edges also need to be removed. The 2x2 Hessian matrix ($H$) is used to compute the principal curvature ratio. If this ratio is greater than a threshold, that keypoint is discarded. It allows to eliminate any low-contrast keypoints or edge keypoints what remains only strong interest points.

- Orientation assignment

To achieve invariance to image rotation an orientation must be assigned to each keypoint. A neighborhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region.

For each image sample $L(x, y)$ the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ can be calculated as:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (20)$$

$$\theta(x,y) = \tan^{-1}\left((L(x,y+1) - L(x,y-1)) / (L(x+1,y) - L(x-1,y))\right) \quad (21)$$

Based on original source (Lowe 2004): "An orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint. The orientation histogram has 36 bins covering the 360 degree range of orientations. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a $\sigma$ that is 1.5 times that of the scale of the keypoint.

Peaks in the orientation histogram correspond to dominant directions of local gradients.

The highest peak in the histogram is detected, and then any other local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation. Therefore, for locations with multiple peaks of similar magnitude, there will be multiple keypoints created at the same location and scale but different orientations."

The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. It creates keypoints with same location and scale, but different directions. It contribute to stability of matching.

- Key point descriptor

When the keypoint descriptor is created a 16x16 neighborhood around the keypoint can be taken and divided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histogram is created and total of 128 bin values are available. It is represented as a vector to form keypoint descriptor (*OpenCV Tutorials* 2015). Key points between two images can be matched by using their nearest neighbors. KNN algorithm can be used for search to locate close matches for further classification. The quantity of features is particularly important for object recognition, where the ability to detect small objects in cluttered backgrounds requires that at least three features be correctly matched from each object for reliable identification.

# 4 EXPERIMENTS AND RESULTS

## 4.1 Data acquisition and pre-processing

The two types of video sequences are used as an initial materials for this work. The first type is synthetic video sequence, which was made in non-real nature situation, for example, the video of the fish in aquarium. This type of data of a very good quality and it almost impossible to obtain in real nature. This video can be used for development of methods of the research. For this research work the synthetic video example was made in Kotka Maretarium, Finland (address: Sapokankatu 2, 48100 Kotka). The second type consist of the materials which were made in real environment by organization "Kymijoen vesi ja ympäristö ry" (address: Tapiontie 2 C, 45160 Kouvola) in 2013 (Figure 16). In this case there are a lot of difficulties refering to specific type of lighting, reflection and color reproduction. This type of video sequences are used for finding and subsequent tuning of parameters, testing. For real type video it is reasonable to apply the background subtraction (BS) methods.

**Figure 16.** Real (left) and synthetic (right) video sequence examples.

The problem of the recognition of fish from real video is also simplified by the fact that the fish in the considered real video material always moves upstream, from right to left. Meanwhile, irrelevant objects are always moving from left to the right side. In this case it is possible to determine a fish and calculate the amount of fish by tracking the objects.

The scheme of camera position and fish position based on the information which was received from "Kymijoen vesi ja ympäristö ry" are presented in Figure 17.

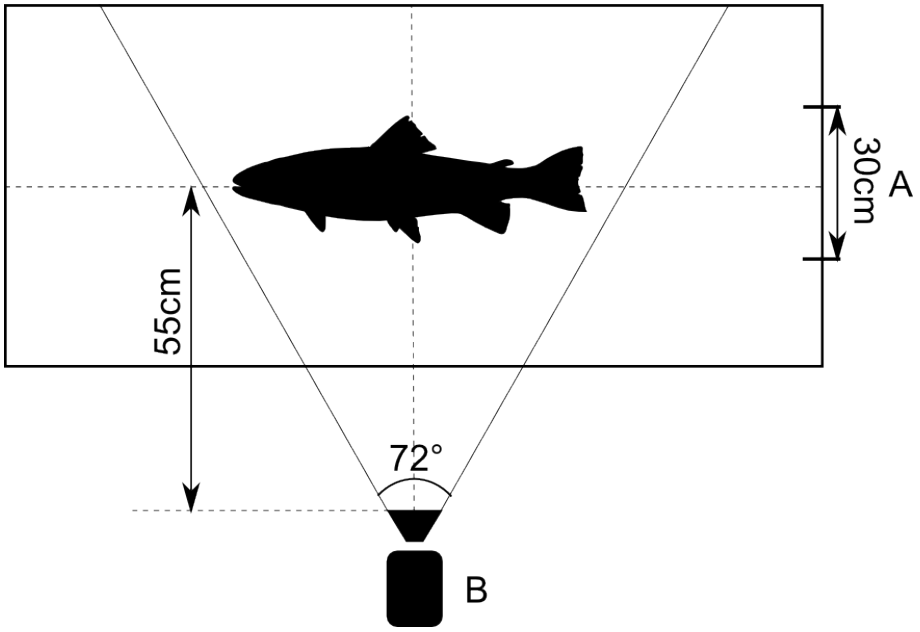



**Figure 17.** Scheme of the scene and camera position for real type video.

Scheme of the camera position for real video type . The point A is the entering place, through which the fish enter to the tube. The width of the entrance is 30 cm. The point B is the camera position. The angle of camera view is 72°. The mean value of the distance between fish and camera is 55 cm. But the position of fish related to the camera cannot be estimated based on available information. This fact complicates the procedure of fish size estimation. For the synthetic video the conditions are different. Although the synthetic video is a better quality and any fish from this video can be easily recognized for human, it is still a big problem for computer vision system because of light reflection of water and aquarium front glass. Moreover, the testing samples contain a big amount of fish in the scene. It makes difficult the using such methods as e.g. background subtraction method. Also in this case the fish cannot be calculated correctly because of overlapping and noise (the tracking task cannot be executed satisfactory).

## 4.2 Implementation of the methods

The experimental part of the research was conducted on different datasets. This is due to the fact that with the data obtained in real conditions there are often problems with using of a particular method because it is impossible to get good results for poor image quality. At the same time, this fact does not mean that the methods are not applicable to all similar data set. The same method can be successfully used with test data of higher quality, for

example. In the program realization the methods which were described in chapter two are implemented. The OpenCV library are used. Initially the research problem was divided into several subtasks, namely:

- Object detection.
- Object tracking.
- Object classification and object size estimation.

In perfect conditions the implemented system should work with real type video. But in real life, as it was indicated above, it is impossible to get good results for real video sequence. In this case some methods were implemented for synthetic video and then tested with real video. The goal of this approach is to show that results can be achieved for the similar detection target if the quality of the video will be improved in future.

In the first experiment with real type video the implemented solution consists of two algorithms: fish detector and fish tracker. Fish detector bases on Gaussian mixture model background subtraction. Background subtraction is a common and widely used technique for generating a foreground mask (namely, a binary image containing the pixels belonging to moving objects in the scene) by using static cameras. The real video has static background. This fact make suitable the implementation of such methods as background subtraction method, because there is no problem with getting original background image from video sequences.

Main steps in the "Fish detector" algorithm:

1. Construct the background.
2. Get next frame from the video sequence and perform background subtraction on it. The result of subtraction performs a foreground mask.
3. Enhance the mask by applying morphological operations to remove noise and fill in holes.
4. Perform a blob detection.
5. Filter small objects by using area threshold (threshold value was chosen experimentally).
6. Find the bounding boxes for all remaining objects.

7. Pass the result to the "Fish tracker" algorithm.

Fish tracker module uses Kalman filter for tracking objects. The filter can be used to predict the real position of something being tracked at a better accuracy than raw sensor data. The Kalman filter uses the history of measurements to build a model of the state of the system that maximizes the probability for the position of the target based on the past measurements. Kalman filter set up with 4 dynamic parameters and 2 measurement parameters (no control), where measurement is: 2D location of object, and dynamic is: 2D location and 2D velocity.

Since Kalman filter is a iterative estimator, it needs only the estimated state from the previous time step and the current measurement to compute the estimate for the current state. In contrast to batch estimation techniques, no history of observations and/or estimates is required. This can be very helpful to improve tracking of the objects.

Steps in the "Fish tracker" algorithm:

1. Receive detected objects from Fish detector module.
2. Predict position by using Kalman filter for each tracked object (fish).
3. Find a corresponding detected objects by comparing distance between predicted and detected position.
4. Correct a Kalman filter with real values.
5. Mark a remaining unmatched fishes for the list of lost fishes.
6. Add remaining unmatched objects into the list of traced fishes and assign an identifications to the each of them.
7. Increase age value of lost fishes and remove old, which have age value more than threshold.
8. Display detected fishes.

## 4.3 Results of the experiments

The results of the experiments using background subtraction method and Kalman filtering are shown in Figures 18 and 19.


**Figure 18.** The result of the experiment with real video.

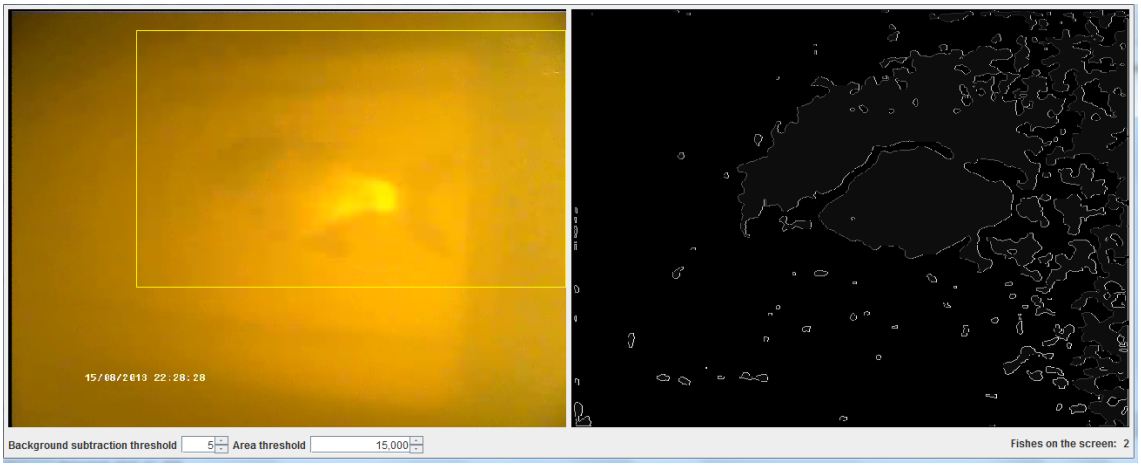



**Figure 19.** The result of the experiment with real video. Threshold value = 5.

Background subtraction is sensitive to changing illumination and unimportant movement of the background (such as reflections of sunlight or water in case of the during experiments). Often the choice of threshold value for frame differencing depends on the size and speed of the object.

As seen from the first experiment, the result contains a lot of noise. This is mainly due to the reflection of light rays from the water flow. In this case the threshold value is minimum. The contours of the object of interest (a fish) in this case cannot be clearly distinguished from the video frame. The rectangle that is drawn around the object has a visible size much larger than the size of the object of interest (Figure 19), since the area of the object is considered to be wrong because of the noise of the image. For that reason

the threshold value of the absolute difference to get the foreground mask was increased in the next experiment. The result of the second experiment is shown in Figures 20 and 21.

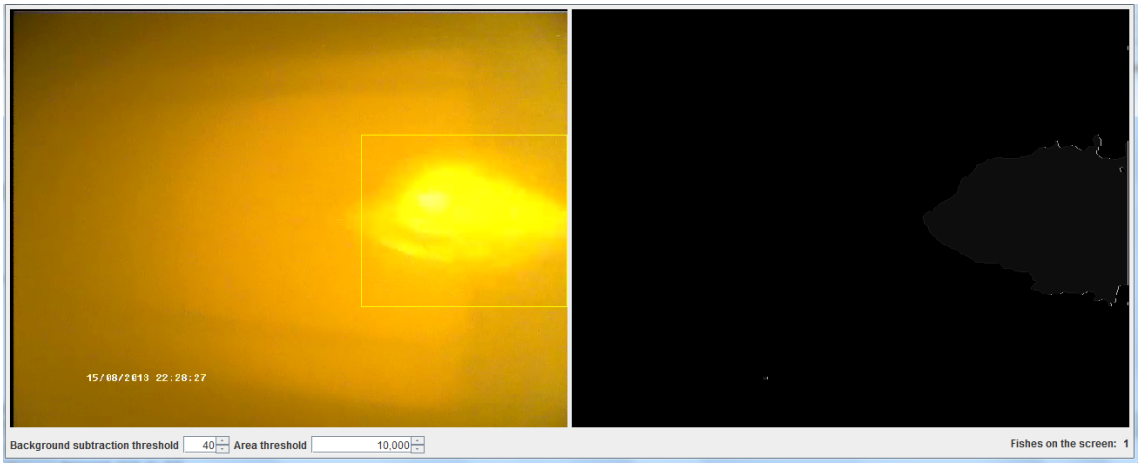



**Figure 20.** The result of the second experiment with real video. Threshold value = 40.

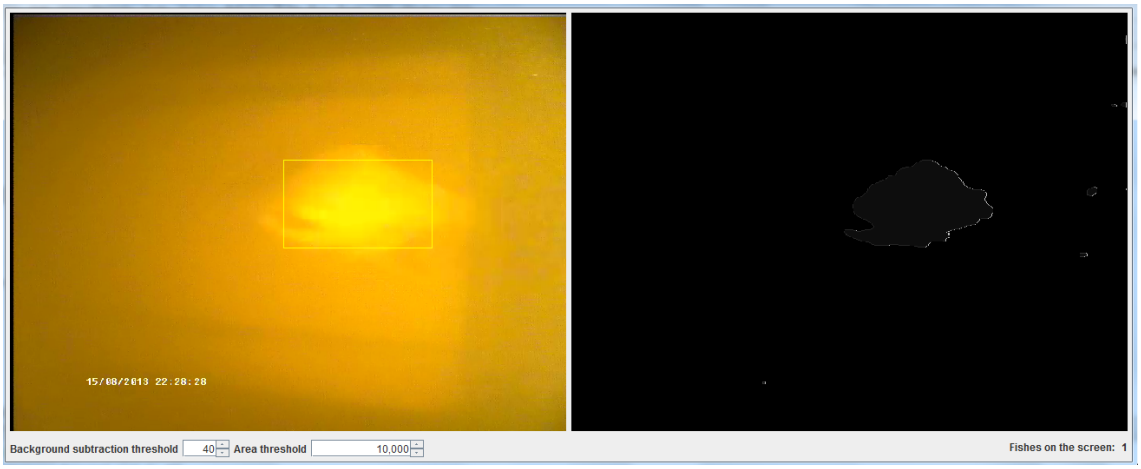



**Figure 21.** The result of the second experiment with real video. Threshold value = 40.

Threshold value for second experiment was increased from 8 to 40. The results show that in this case the object boundaries can be much better estimated from the video. But the problems relating to the image quality are still presented. In some frames the color of object is exactly the same as a background color, mainly because light reflection. In this situation one object can be found as two different objects and object area can be calculated incorrect. Figure 22 represents a result of the experiment with wrong object detection case. As we can see, the rectangle is drawn around part of fish, which was detected by the algorithm as an independent object.


**Figure 22.** The problem in experiments with BS method (real video).

A similar experiment was performed with synthetic video, but any acceptable results have not been reached (Figure 23). Nevertheless, experiment with background subtraction technique showed very good result for real video type.

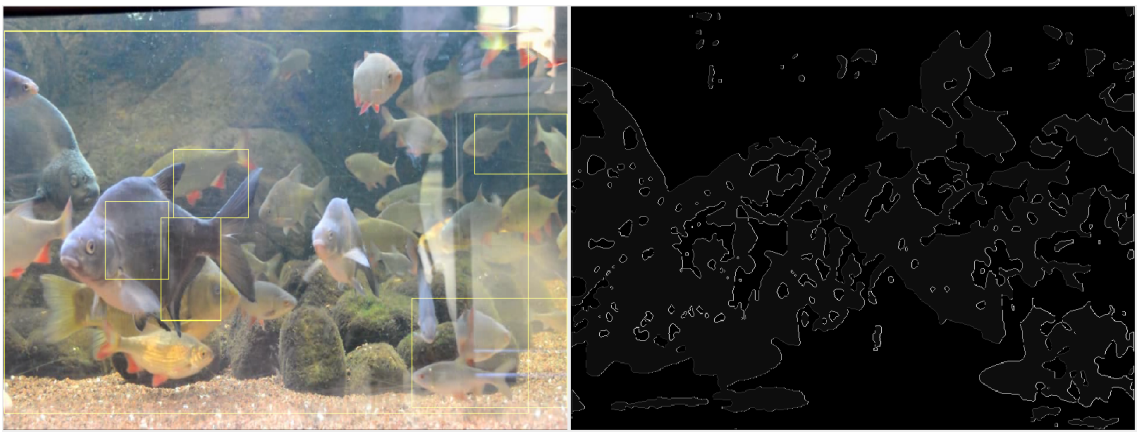


**Figure 23.** The problem in experiments with BS method (synthetic video).

The next experiment was performed with synthetic video sequence. The SIFT algorithm was selected for fish detection in this video because it show good performance in similar applications (Zeppelzauer, 2013, Ramanan et al., 2006). Following steps were executed during the realization:

- Perform a feature extraction on the reference image of the fish (Figure 24).
- Extract a frame from the video source.

- Perform a feature extraction on the extracted frame.
- Perform a matching between the reference keypoints and frame keypoints by using kNN algorithm.
- Filter false positive matching using threshold on distance between keypoints.
- Find bounding box for good matches.

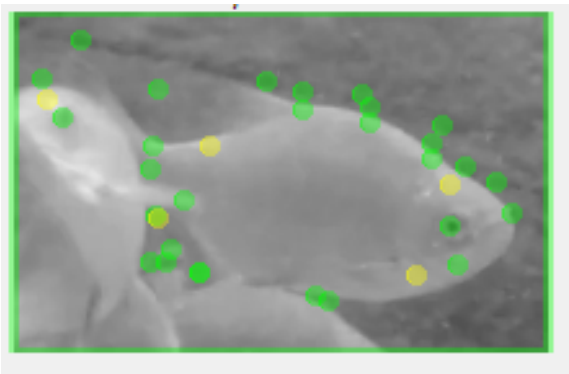


**Figure 24.** Example of the detected keypoints on the reference image.

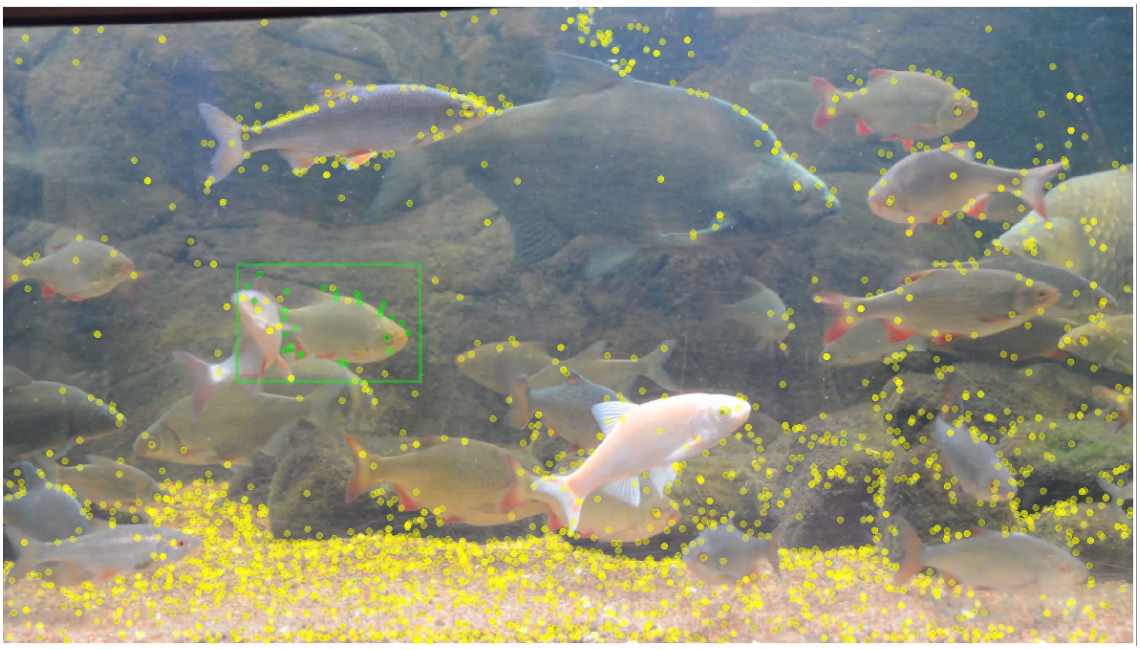


**Figure 25.** Example of the detected matches on captured frame.

Unfortunately this method did not show any satisfactory results for synthetic testing data.

The result of the experiment are shown in Figure 26. The main reason for the poor quality of the results is a low contrast of image frames and therefore in practice it is impossible to find a correspondence between features because of similarity. This problem is relevant to the both of video types.

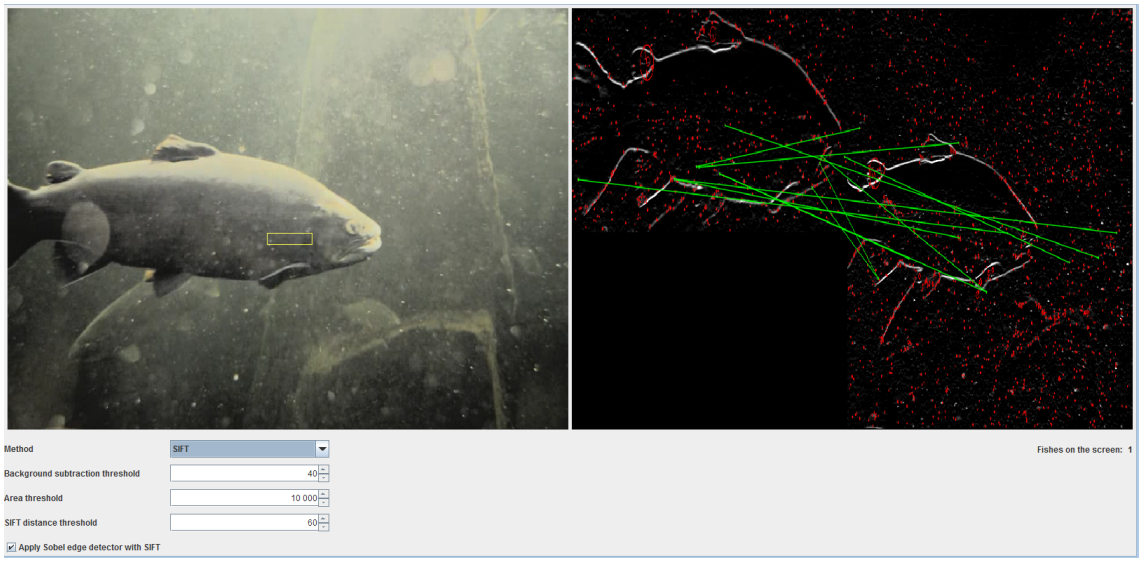



**Figure 26.** Result of the experiment with using SIFT method.

During the work of algorithm, the image features were not be matched correctly for any object in scene. Usually the algorithm found certain keypoints, but in each frame these points are different, which make it difficult to find the object of interest. Quality and quantity of points found was also insufficient. Basically the reason is a changing in lighting model due to the glare of water in the aquarium and water turbidity.

The last experiment was organized with synthetic video sequence by implementation Haar cascades. In this method a classifier which is called a cascade of boosted classifiers and works with haar-like features is trained with a few sample views of a fish, (positive examples) and with arbitrary images of the same size (negative examples). Figures 27 and 28 show the examples of positive and negative images for real and synthetic video types.


**Figure 27.** Positive (left) and negative (right) examples for classifier training (real video).

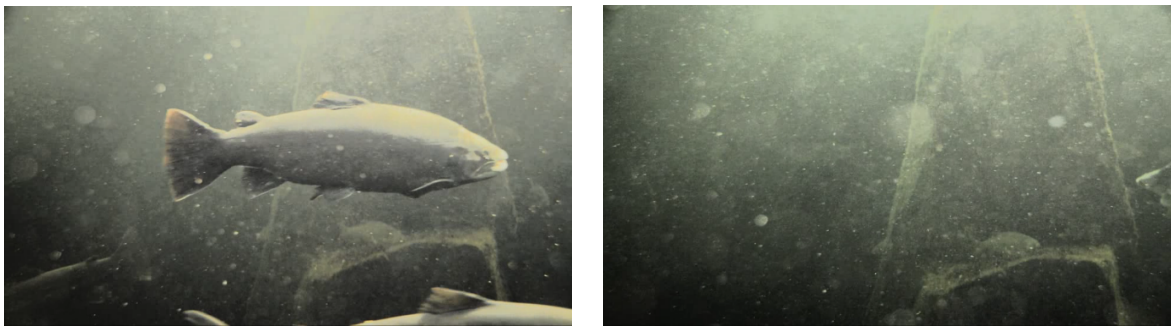


**Figure 28.** Positive (left) and negative (right) examples for classifier training (synthetic video).

When the classifier is trained, it can be applied to a region of interest (of the same size as used during the training) in an input image. The classifier outputs is "1" if the region is likely to show the object, and "0" otherwise. To search for the object in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily "resized" in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. Finding of an object of an unknown size in the image the scan procedure should be done several times at different scales (*OpenCV Tutorials* 2015).

Figure 29 represents the results of Haar cascades method. The word "cascade" in the classifier name means that the resultant classifier consists of several simpler classifiers (stages) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. The word "boosted" means that the classi-

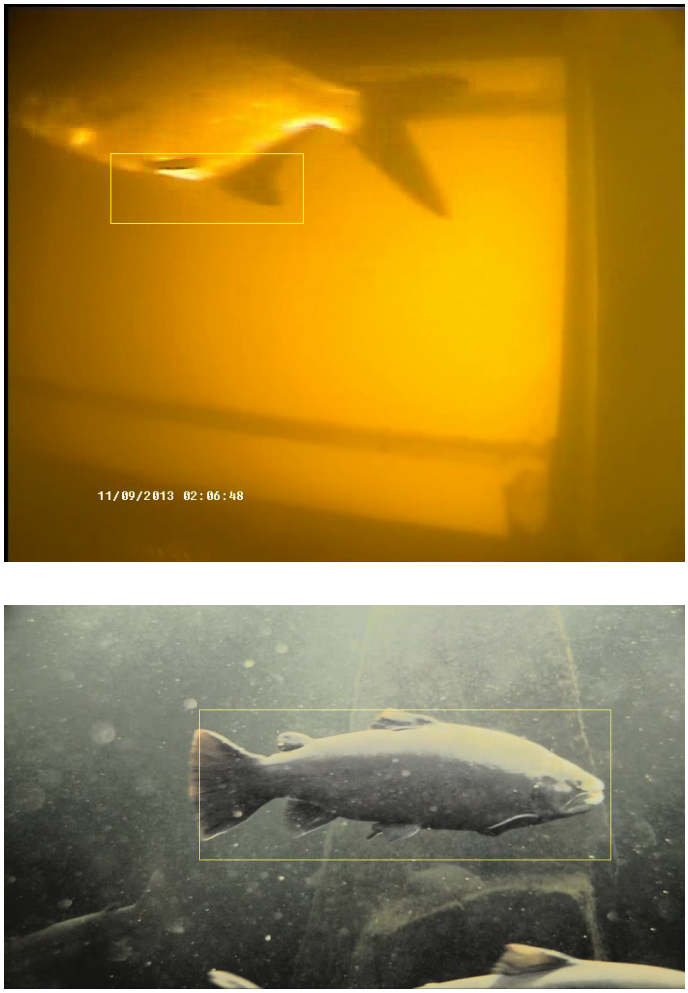



**Figure 29.** The results with using Haar cascade.

fiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques (weighted voting). The basic classifiers are decision-tree classifiers with at least 2 leaves (*OpenCV Tutorials* 2015). Haar-like features are the input to the basic classifiers and are calculated as described below.

## 4.4 Comparison of the approaches

Table 1 demonstrates a compared results of using methods which were described above. Several video sequences of both types were used in the experiments. The column "Total frames" contains total number of frames in each video sequence (also represented in percents). The other columns contain the results of implementation indicated methods for each sample. Last two rows represent the mean value for real and synthetic video type

separately for each method. The accuracy is calculated as a ratio of the number of frames in the testing video sequence to the number of frames in which the fish was successfully recognized:

$$acc = \frac{F_t}{F_d} * 100\% \tag{22}$$

where $F_t$ - total number of frames in video sequence which contain object of interest, $F_d$ - number of frames in which object of interest was successfully recognized. In the context of the presented experiment the successfully treated frame means a frame in which the object of interest (a fish) was detected correctly (a boundary box containing more than 60% of the object area). For example, see Figures 21, 29.

**Table 1.** Results of the experiments

| Type of video | Total frames | BS | SIFT | Haar Cascades |
|---|---|---|---|---|
| Video 1 (real) | 204 / 100% | 145 / 71.1% | N/A | 0 / 0% |
| Video 2 (real) | 22 / 100% | 18 / 81.8% | N/A | 0 / 0% |
| Video 3 (real) | 26 / 100% | 21 / 80.7% | N/A | 5 / 19.2% |
| Video 1 (Synthetic) | 559 / 100% | 0 / 0% | 5 / 0.01% | 398 / 71.2% |
| Video 2 (Synthetic) | 192 / 100% | 0 / 0% | 19 / 9.8% | 105 / 54.6% |
| mean for real type | - | 73% | 0 | 2% |
| mean for Synthetic type | - | 0 % | 3.2% | 67% |

Table 2 represents the number of processed frames per second for each methods.

**Table 2.** Results of the experiments (frame per second score).

| Type of video | BS | SIFT | Haar Cascades |
|---|---|---|---|
| Real | 12.9 | N/A | 18.3 |
| Synthetic | 9.7 | 1.55 | 13.3 |

The results of the experiments show that for the real video type the highest effectiveness was received by using the background subtraction method. Whereas for the synthetic video sequence the method of Viola-Jones with Haar cascades showed the best result. Nevertheless in both cases the maximum frame per second ratio show method of Viola-Jones. This parameter is especially important for task of object recognition in real-time.

# 5 DISCUSSION

## 5.1 Discussion of the results

In this section the results of the experiments were considered. Based on the available test data it is possible to conclude that the problem of detecting and tracking objects can be completed using the described methods. The problem of the fish size and type estimation in this case is more complicated by the following facts:

- Inability to determine the exact distance to the object.
- The lack of information about the angle of inclination of the object relative to the viewpoint.
- Inability to determine the precise boundaries of the object because of the large amount of noise which impossible completely to get rid of (too much computational errors even for "successful" frames, hence there are doubts in the objectivity of the results).

The possible ways of solving problems related to the parameters such as a more high definition image, adjusting lighting model. It is also possible to use additional equipment to determine the exact position of an object in the monitored area. The classification problem has similar challenges to implementation. However, in the case of using a Haar cascade with the synthetic test data it was impossible to get a good set of a negative image examples because of the large number of objects in the scene. This fact makes it impossible to train the classifier with some data sets. Unfortunately, in this research work only limited set of test samples (examples) for the fish size estimation are available.

## 5.2 Future work

The subject of automatic fish recognition system has wide field for further work. There are several ways for improving exist solution, but in the same time the technical parameters of the system also affect to the results. In practice, it is always easier to use camera with higher resolution than to implement more complex methods for solving of the similar problem. From this point of view, the technical parameters of the system should be improved for getting better video quality. In this case the accuracy of fish classification

may be improved a lot. The methods Viola-Jones using Haar cascades and background subtraction may be used for real video sequences.

# 6 CONCLUSION

In the current work the methods of detection and tracking of fish in the water were considered. The results of the experiments show that for this problem the quality of the used video sequence plays the decisive role. The results of the experiments which are presented in Table 1 can be interpreted as a comparative analysis of different methods for certain types of source data sets. The most accurate results (about 73% frames from video sequence were recognized correctly) for the real type video have been obtained using the method of the background subtraction. Meanwhile, for the synthetic video sequence the best result is achieved by implementation of the method Viola-Jones using Haar cascades. As can be assumed of the given results, some methods may not give satisfactory results because the data has a large number of image noise. In the other cases, the techniques which were described in this Master's Thesis may be successfully applied to achieve the goals.

# REFERENCES


Arnarson, H., Bengoetxea, K., and Pau, L. (1988). "Vision applications in the fishing and fish product industries". In: *Int J Pattern Recognition Artificial Intelligence*, no. 2, pp. 657–671.

Athanesious, J. and Suresh, P. (2012). "Systematic Survey on Object Tracking Methods in Video". In: *International Journal of Advanced Research in Computer Engineering & Technology*, pp. 242–247.

Bovik, A. (2005). Handbook of Image and Video Processing (Communications, Networking and Multimedia). Orlando, FL, USA: Academic Press, Inc.;

Christiansen, P., Steen, K. A., Jørgensen, R. N., and Karstoft, H. (2014). "Automated Detection and Recognition of Wildlife Using Thermal Cameras". In: *Sensors* vol. 14, no. 8, pp. 13778–13793.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). Pattern Classification (2nd Edition). Wiley-Interscience;

Faragher, R. (2012). "Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation [Lecture Notes]." In: *IEEE Signal Processing Magazine* vol. 29, no. 5, pp. 128–132.

Frantz, V., Voronin, V., Marchuk, V., Fisunov, A., and Pismenskova, M. (2013). "The algorithm for constructing the trajectories of moving objects in a video based on optical flow". In: *Engineering Journal of Don*, no. 3, pp. 1–9.

Gonzalez, R. C., Woods, R. E., and Eddins, S. L. (2003). Digital Image Processing Using MATLAB. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.;

Grewal, M. S. and Andrews, A. P. (2001). Kalman filtering : theory and practice using MATLAB. New York, Chicester, Weinheim: Wiley;

Guo, Z. (2001). Object Detection and Tracking in Video. Tech. rep. Department of Computer Science, Kent State University.

Hartley, R. and Zisserman, A. (2004). Multiple View Geometry in Computer Vision. Second. Cambridge University Press, ISBN: 0521540518;

Horprasert, T., Harwood, D., and Davis, L. S. (1999). “A statistical approach for real-time robust background subtraction and shadow detection”. In: pp. 1–19.

Kaewtrakulpong, P. and Bowden, R. (2001). An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection.

Kalman, R. E. (1960). “A New Approach to Linear Filtering And Prediction Problems”. In: *ASME Journal of Basic Engineering*, pp. 35–45.

Lee, J.-Y. and Yu, W. (2011). “Visual tracking by partition-based histogram backprojection and maximum support criteria.” In: *ROBIO*. IEEE; pp. 2860–2865.

Lowe, D. G. (2004). “Distinctive Image Features from Scale-Invariant Keypoints”. In: *Int. J. Comput. Vision* vol. 60, no. 2, pp. 91–110.

Maggio, E. and Cavallaro, A. (2011). Video Tracking, Theory and Practice. Pearson, Prentice Hall;

Mathiassen, J. R., Jansson, S., Veliyulin, E., Njaa, T., Lønseth, M., Bondø, M., Østvik, S. O., Risdal, J., and Skavhaug, A. (2006). “Automatic weight and quality grading of whole pelagic fish”. In: *Nor-Fishing Technology Conference*, pp. 1–8.

Misimi, E., Mathiassen, J. R., and Erikson, U. (2007). “Computer vision-based sorting of Atlantic salmon (Salmo salar) fillets according to their color level”. In: *Journal of food science*, pp. 30–35.

OpenCV Tutorials (2015). [online document]. [Accessed: 18 May 2015 ]. Available at: `http://docs.opencv.org/doc/tutorials/tutorials.html`.

Parekh, H. S., Thakore, D. G., and Jaliya, U. K. (2014). “A Survey on Object Detection and Tracking Methods”. In: *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)* vol. 2, no. 2, pp. 2970–2978.

Ramanan, D., Forsyth, D. A., and Barnard, K. (2006). “Building models of animals from video”. In: *IEEE Trans Pattern Anal Mach Intell* vol. 28, no. 8, pp. 1319–1334.

Simard, P. Y., Bottou, L., Haffner, P., and LeCun, Y. (1999). “Boxlets: A Fast Convolution Algorithm for Signal Processing and Neural Networks”. In: *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*. Cambridge, MA, USA: MIT Press; pp. 571–577.

Spampinato, C. and Palazzo, S. (2012). “Hidden Markov Models For Detecting Anomalous Fish Trajectories In Underwater Footage”. In: *Proceedings of the 2012 IEEE International Workshop on Machine Learning for Signal Processing*. Santander, SPAIN: IEEE Computer Society; pp. 23–26.

Spampinato, C., Giordano, D., Di Salvo, R., Chen-Burger, Y.-H., Fisher, R. B., and Nadarajan, G. (2010). “Automatic Fish Classification for Underwater Species Behavior Understanding”. In: *Proceedings of the First ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*. ARTEMIS ’10. New York, NY, USA: ACM; pp. 45–50.

Suzuki, N., Hirasawa, K., Tanaka, K., Kobayashi, Y., Sato, Y., and Fujino, Y. (2007). “Learning motion patterns and anomaly detection by Human trajectory analysis.” In: *SMC*. IEEE; pp. 498–503.

Viola, P. and Jones, M. (2001). “Rapid object detection using a boosted cascade of simple features”. In: pp. 511–518.

Viola, P. and Jones, M. (2004). “Robust Real-Time Face Detection”. In: *Int. J. Comput. Vision* vol. 57, no. 2, pp. 137–154.

Welch, G. and Bishop, G. (1995). An Introduction to the Kalman Filter. Tech. rep. Chapel Hill, NC, USA.

Zeppelzauer, M. (2013). “Automated detection of elephants in wildlife video”. In: *EURASIP Journal on Image and Video Processing* vol. 2013, no. 46, pp. 1–23.