Robot Autonomous Fire Location using a Weighted Probability Algorithm

Chris Lorena Nogales

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

A. Lynn Abbott, Chair
Brian Y. Lattimer
Pratap Tokekar

August 29, 2016
Blacksburg, Virginia

Keywords: autonomy, perception, machine learning, firefighting robot

Robot Autonomous Fire Location using a Weighted Probability Algorithm

Chris Lorena Nogales

(ABSTRACT ACADEMIC)

Locating a fire inside of a structure with no map of the environment or sensor information about conditions poses a dangerous threat to the safety of firefighters. As a result, robots are being explored to increase situational awareness inside structures prior to a firefighter entering. This thesis presents an algorithm that autonomously guides a mobile robot to the location of a fire by determining a heading based on the highest weighted probability region in an IR image. This heading is compared with a heading determined from a maximum temperature algorithm using the IR image. Experimental results have demonstrated the validity of the proposed algorithms for a series of large-scale fire tests with a robot moving toward a fire.

Robot Autonomous Fire Location using a Weighted Probability Algorithm

Chris Lorena Nogales

(ABSTRACT PUBLIC)

Finding a fire inside of a structure without knowing its conditions poses a dangerous threat to the safety of firefighters. As a result, robots are being explored to increase awareness of the conditions inside structures before having firefighter enter. This thesis presents a method that autonomously guides a robot to the location of a fire inside a structure. The method uses classification of fire, smoke, and other fire environment objects to calculate a weighted probability. Weighted probability is a measurement that indicates the probability that a given region on an infra-red image will lead to fire. This method was tested on large-scale fire videos with a robot moving towards a fire and it is also compared to following the highest temperatures on the image. Sending a robot to find a fire has the potential to save the lives of firefighters.

Dedication

This thesis is dedicated to my grandparents, Florencia and Jose Iturri, who taught me laughter, optimism, and hard-work.

Acknowledgments

This work was sponsored by the Office of Naval Research through Grant N00014-15-1-2128.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Locating a fire inside of a structure with no map of the environment or sensor information about conditions poses a serious threat to the safety of firefighters. As a result, robots are being explored to increase situational awareness inside structures prior to a firefighter entering. This thesis presents an algorithm that autonomously guides a mobile robot to the location of a fire. In this research, this was accomplished by classification of the fire environment and using class features to guide a robot.

1.1 Motivation

As best quoted from the International Fire Service Training Association by Shelley et al. [1] "Of all the perils at sea, one of the most frightening is fire. Difficult to deal with and devastating in its effects, fire at sea leaves the mariner caught between two unforgiving elements".

The main motivation for this research is to reduce the risk of death for firefighters. According to the most recent NFPA report on Firefighter Fatalities in the United States, out of the 64 firefighters that died in 2014, 22 died while operating at fire events. From these fatalities, 17 out of the 22 occurred in dwellings and enclosed properties [2]. Firefighters died because of structural collapse, getting trapped or caught by rapid fire progress, getting lost inside the structures and inhaling smoke or asphyxiating. Having a robot, or any other similar mobile platform, serve as fire emergency responders would help decrease firefighter casualties. Although this research began specifically as a part of the SAFFiR (Shipboard Autonomous Firefighting Robot) project sponsored by the ONR (Office of Naval Research) for ships, the goals remains the same for structures. The purpose of this research is to create an algorithm to autonomously guide a mobile robot to the location of a fire outside its field of view (FOV) to avoid human exposure to fire emergencies.

1.2 Challenges

Rapid spread of fire, low visibility, structural collapse, high temperatures, and low oxygen are dangers inherent to confined spaces that are only increased in shipboard firefighting. Sending a robot deep into the hull of a vessel would ease the severity of these dangers because loss of the robot would be equivalent to loss of property. However, a shipboard firefighting environment poses a different set of challenges to the perception system than that posed to traditional emergency response robots.

The first challenge is reduced visibility. Current state-of-the-art perception technology shows amazing progress in classification of objects using machine learning techniques. However, these systems mainly use static RGB cameras. RGB cameras would not be able to pick up cues necessary for finding a fire in the dark. A second challenge is high density smoke and its particles. The perception system may correctly classify smoke as such but the smoke may cover the entire FOV of the robot. Hence, finding a way towards the fire would require analysis of the surroundings and human understanding of the implication of fire given that smoke is detected. Finally, one of the most important challenges is having the ability to run the system in real-time in a changing environment. Should the robot camera have to rotate or move in novel configurations as the robot traverses the ship, the perception system will need to account for these movements in real-time.

The main research of this thesis is to provide an algorithm that can autonomously guide a mobile robot to a fire outside its field of view (FOV) in real-time. To do so, a real-time classification based system using a single infra-red (IR) camera was chosen to develop the algorithm in C++ and Robot Operating System (ROS).

1.3 Literature Review

Current firefighting platforms in both research and industry are mainly tele-operated. Limited research has been performed on autonomous location of fires that are outside the FOV of a robot, with the majority of robots being remote controlled and relying on the operator to make decisions on the robot heading. Additionally, state-of-the art techniques for identification of fire and smoke mainly use static RGB cameras with the goal of early warning.

1.3.1 How do Firefighters Find Fires?

Typical initial search operations firefighters use during a fire event are divided into two types: primary and secondary search. It is during the primary search that firefighters prioritize quickly searching for life and fire. One of the main tools firefighters use to find fires are

hand-held thermal-imaging devices to see through smoke, measure temperatures, and help guide them to the highest temperatures. In order keep fires at bay and find their way back, first responders also go inside a building with a hose line. It is important to identify the location of the fire in order to prioritize areas in need of search and work away from the fire. However, when their thermal imaging tools fail they rely on coordinated and land-marked searches [3] [4].

1.3.2 Robotics and Mechanisms in Firefighting

There are several robots that have been designed to locate and suppress fires, and several methods are used to find fires. While the ultimate goal for robotics in firefighting is to have autonomy, previous work shows that robots that put out fires rely on tele-operation from humans.

Commercial robots for firefighting are usually threaded platforms with hoses fitted on them. These platforms are tele-operated to get into an appropriate range in order to put out the fire. Shown in Figure 1.1 are six unmanned commercial robots used around the world.

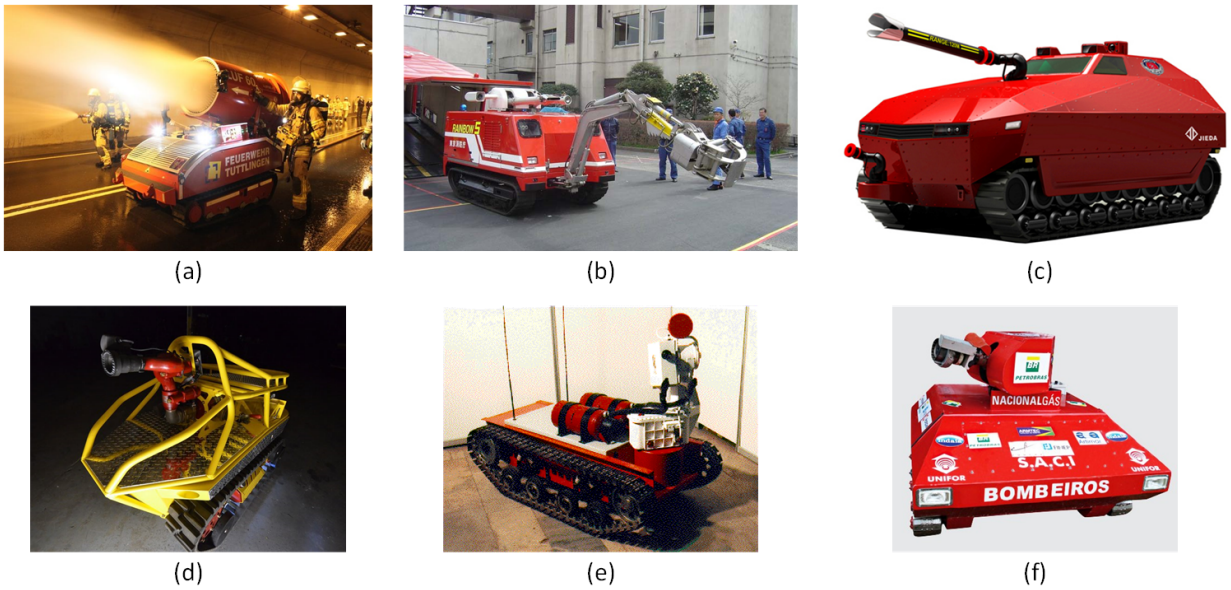

Figure 1.1: (a) Luf–60 (b) Rainbow5 (c) "Large Fire Fighting Robot" (d) Thermite (e) CNII-RTC "Fireman" (f) SACI.

Figure 1.1 (a) shows Luf–60 is a wirelessly controlled German robot developed by Rechners GmbH in Austria capable of climbing stairs and shooting water or foam 60 meters away [5] [6]. Figure 1.1 (b) shows Rainbow5, developed by the Tokyo Fire Department, a tele-operated platform designed to extinguish unapproachable large fires such as aircarft fires [5]. Figure 1.1 (c) presents "Large Fire Fighting Robot" Fire Truck by JIEDA Fire-Protection in China.

Figure 1.1 (d) presents a smaller remote-controlled platform called Thermite, developed by Howe and Howe Technologies in U.S.A., a robotic platform that can be tele-operated from 400 meters away [7]. Figure Figure 1.1 (e) shows tele-operated robot "Fireman" build by CNII-RTC in Russia [5] [8]. Finally, Figure 1.1 (f) shows SACI 2.0, a robot developed in Brazil by ARMTEC. This platform can be controlled either wirelessly or by cable [9].

Figure 1.2 shows research platforms that follow the same tele-operation trend as the commercial platforms but tend to be smaller [10] [11] [12] [13] [14]. Most of the research platforms for firefighting also rely on tele-operation from humans. KAIST university in Korea developed a remote controlled robot for indoor underground fires [10]. Another relevant small remote-controlled robot goes into a fire environment for pre-mapping and communication; this robot was tested against high temperatures with the goals of portability and warning of flashover [12]. Such small platforms are designed with the goals of exploration, communication, and warning but the robotic platforms in this research also have the goal of fire suppression.

Despite the small form factor, there are three research platforms that are most similar to this research. First, Figure 1.2 (a) shows a line following robot called AFFMP (Autonomous Fire Fighting Mobile Platform). The robot follows a line to cover an area and only changes behavior if its flame sensor detects a small flame [11]. Second, Figure 1.2 (b) shows another small custom-made robot that uses x-ray films to reduce false detection of fires and computer vision with thresholding heuristics. The robot aligns itself to the pixel mapping location of the fire in its FOV [14]. Third, a small LEGO robot uses a voting logic algorithm on several sensors to guide itself to a fire. The small robot moves in a sinusoidal fashion as it traverses in order to find a fire [13].

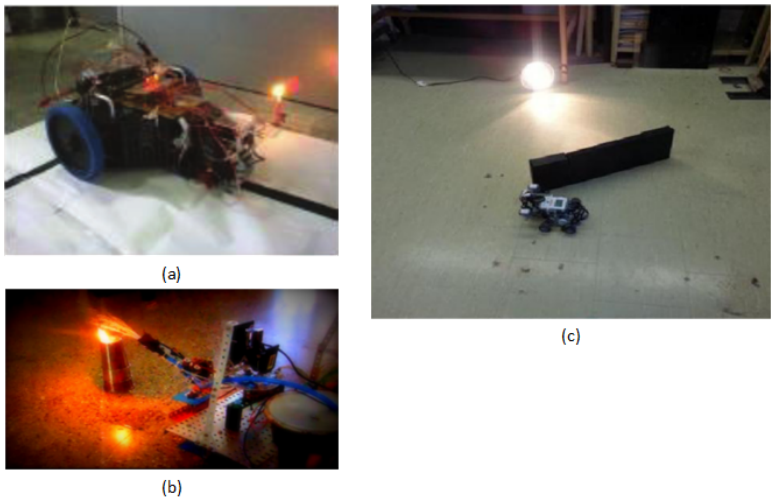

Figure 1.2: (a) Line following robot AFFMP (image credit [11]) (b) An unnamed custom robot moves in a sinusoidal fashion (image credit [14] ©[2013]$IEEE$) (c) LEGO robot uses voting logic (image credit [13] ©[2014]$IEEE$)).

In the first approach, the robot takes on a surveillance role but shipboard personnel would require extra material to cover a floor plan with a line. The use of x-ray film on RGB video is interesting but obscurity would impede usability in ships. In the latter two cases, the robots could also take on a surveillance role but would respond to the fire alone.

1.3.3 Methods for Identifying Fire and Smoke

Traditional methods of fire warning, such as temperature and particle sampling, have the disadvantage of requiring close proximity for detection [15]. While early computer vision approaches detected fire flames only [16], recent investigations have shifted to detection of both fire and smoke using machine learning and computer vision.

Most methodologies that rely on computer vision and machine learning approaches have the goal of early detection and warning of fire and/or smoke for surveillance, see Table 1.1. Furthermore, the most common sensor used is a color camera to detect fire, smoke, or both in a static Field of View (FOV). Most of these methods want to leverage the already available surveillance cameras in order to detect fire early.

Table 1.1: Chronological literature review of recent machine learning algorithms used for detection of fire and smoke.

Source	Year	ML approach	Detection	Computer Vision Features
[17]	2015	Bayesian	Fire + Smoke + Reflections	mean, variance, inverse difference moment, entropy
[16]	2015	n/a	Fire + Smoke	color, motion
[18]	2014	n/a	Smoke	spatial data
[15]	2013	NN	Smoke	color, diffusion, texture
[19]	2013	SVM	Fire	color, optical flow, orientation
[20]	2012	SVM	Smoke	color, motion
[21]	2011	HMM	Fire	spatial, temporal data
[22]	2011	SVM	Smoke	luminance, texture, temporal data
[23]	2010	NN	Fire + Smoke	wavelets
[24]	2009	SVM	Smoke	color, luminance, temporal data
[25]	2009	Fuzzy Logic	Fire + Smoke	wavelets
[26]	2009	SVM	Fire	spectral, spatial, temporal data
[27]	2008	SVM	Smoke	spatial data

As shown in Table 1.1, common machine learning methods used for early fire warning include back propagated Neural Networks [15] [23], Support Vector Machines (SVM) [19] [20] [22] [24] [26] [27], Hidden Markov Models (HMM) [21], and Fuzzy Logic [25]. Table 1.1 also shows that common computer vision techniques for feature extraction include

wavelets [24] [23], color [16] [15] [19] [20] [24], luminance [22] [24], temporal [21] [22] [24] [26], and spatial features [18] [21] [26] [27]. For all these resources, RGB cameras provide the raw images necessary for the detection algorithms.

A wide variety of computer vision approaches are used to identify fire or smoke. Where a machine learning algorithm is used, computer vision techniques are relied upon for pre-processing or extraction of features that can be fed into the classification algorithms. In cases where no machine learning algorithm is used, thresholding becomes the main decision heuristic but at the expense of having to manually adjust parameters.

Unfortunately, while these methods can reliably detect fire or smoke for this research the localization of a fire by a robot platform imposes certain constraints: dynamic environment from moving camera on robot, obscure environment from dense smoke or possible loss of power, and unreliable wireless communication with the robot.

While many of these applications are meant for indoor use, most would not be useful for a robot searching for fire in obscured environments. This research project requires a method that is real-time, that can work in obscure environments, and that does not rely on a static camera for processing. Therefore, because of the IR ability to image through dense smoke and quantify temperatures IR cameras are used in this research.

1.3.4 Fire Localization Outside the Field of View (FOV) of a Robot

A heuristic algorithm based on maximum temperature from IR images was previously developed by Kim et al. [28] to predict a robot heading for locating a fire; however, it was prone to false headings due to fire reflections or other hot objects. Some research has been conducted to use a Bayesian classification algorithm to distinguish fire, smoke, and reflections [17]. However, these features were not used to assist the robot in making a decision on the robot heading that leads it to the fire. The algorithm was not developed in ROS (Robot Operating System) but instead utilized standard tools for use on the robot.

1.4 Research Objectives

The main objective of this research was to create an algorithm that can guide a robot towards a large fire in an obscured and enclosed indoor environment. In order to provide a base of comparison, the algorithm was compared to a temperature following heuristic. Therefore, the research objectives were

- to develop an algorithm that utilizes classification information in order to guide a robot towards a fire

- to compare algorithm in simulation against a simpler temperature following heuristic

The hypothesis of the approach is that one can subdivide the Regions of Interest (ROI) on the Field of View (FOV) of a camera and use classification information to provide a heading better than temperature following. This research explores further segmentation of classified Regions of Interest (ROIs) and weighting the class of the segmented regions to predict a heading for the robot. The goal of this research is to determine navigation heading information that lead the robot to a fire. The following chapters provide an overview of the research, this includes robot classification, heading approach, results with real fires, simulations, conclusions, and future work.

Chapter 2

Software, Robots, and Simulation

The following chapter presents the details of the localization of fire algorithm, an overview of the robots used in this research, and the software simulation environment. The software section presents a high-level view of the algorithm followed by detailed discussion of each section with images and mathematical background at each stage.

2.1 Software Overview

The localization algorithm was written in C++ and integrated in Robot Operating System (ROS) Indigo with the goal of creating a software stack that can be easily integrated onto other robotics platforms. The following is an overview of the code structure, mathematical background for each sub-module in the algorithm, and the weighted confidence heuristic calculation.

2.1.1 Localization Algorithm Overview

In this research, a new weighted probability heuristic was developed with the intuition that humans use *a priori* knowledge, that is, the presence of smoke indicates the presence of fire. Similarly, if one sees a fire-reflection on a wall then one can assume a fire is nearby. Figure 2.1 contains an overview of the localization algorithm and how the weighted probability measurement aids the robot in determining a final heading angle. This overview includes five images: the original grayscale image, a binary image that includes large ROIs, a winter colormap image that includes sub-ROIs, a visualization of the regions with high weighted probability in green squares, and a masked image showing the final heading.

The localization algorithm can be manually started in a modular manner by calling a launch file in ROS. Once started, the localization algorithm node waits for an input image from the

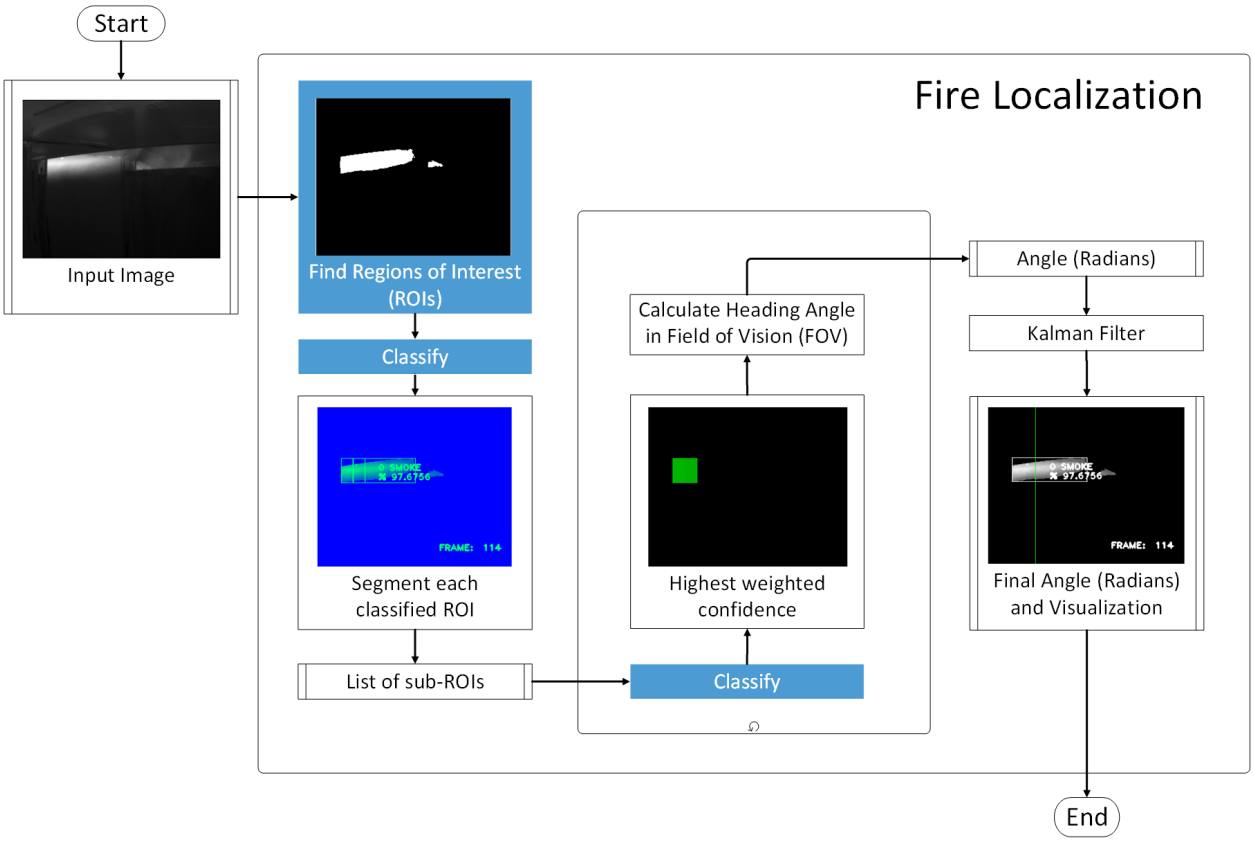

Figure 2.1: Localization algorithm.

IR camera. As shown in Figure 2.1, the grayscale image is pre-processed to find Regions of Interest (ROIs) on the FOV of the image. These large first ROIs are classified and then further segmented. For visualization purposes, a mask and color was added to the original grayscale image using OpenCV's WINTER colormap. The color allows the user to observe fine changes in the grascale images. This blue-green segmentation image also contains squares to indicate the location of the sub-ROIs, segmentation samples are shown in Figure 2.1 and later in Figure 2.5.

In order to choose which sub-ROIs can most likely lead to a fire, the algorithm chooses the region with the highest calculated weighted probability heuristic. Figure 2.1 highlights the sub-ROIs that survived the threshold as green squares. Finally, the algorithm calculates the appropriate angle in order to move the robotic platform with respect to the FOV of the image. A Kalman filter on the angle smooths the signal and the final angle in Radians is sent through ROS in order to move the robot. The highlighted blue boxes in Figure 2.1 will be presented in detail in the next sections.

2.1.2 Finding Regions of Interest

An common and important step when analyzing images in computer vision is segmentation. The goal of segmentation is the extraction of candidate ROIs to remove information that does not pertain to the task at hand. In an IR image, each pixel intensity corresponds to a temperature value. Therefore, agglomerations of high intensity values are chosen for the classification of fire, smoke, and heat-reflection tasks. Figure 2.2 shows an overview of this pre-processing step with corresponding images. Figure 2.2 (a) is the RGB image taken from the entrance of a hallway and Figure 2.2 (b) is the raw IR image of the same entrance. In order to extract ROIs, the pre-processing step starts with Otsu's method of auto-thresholding.

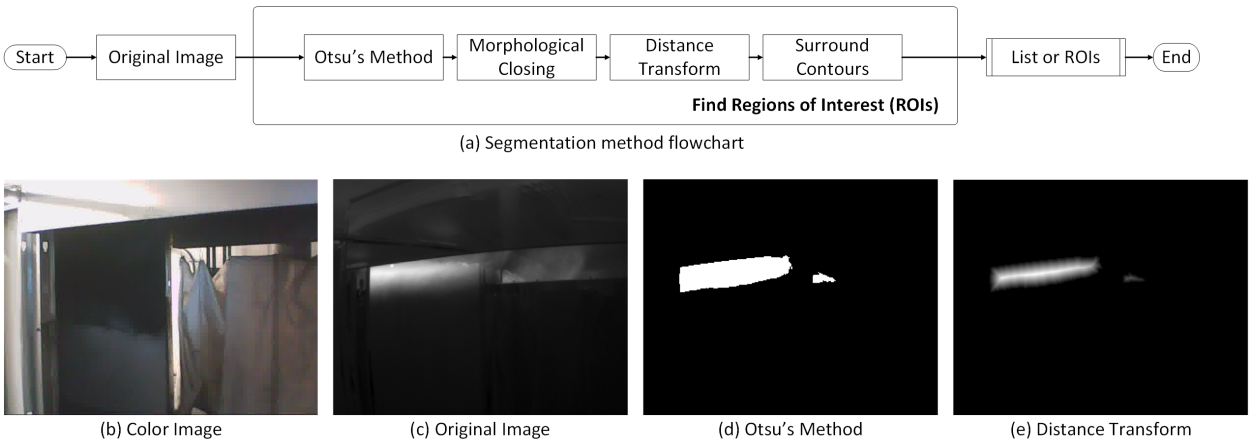

Figure 2.2: Pre-processing sub-routine that finds ROI in the original grayscale image.

2.1.2.1 Otsu's Method

Otsu's method iterates over all possible threshold values that divide a grayscale image's histogram into two classes: a background and a foreground. For each threshold, the method calculates spreads, or variances, of both foreground and background pixel distributions. The optimum threshold is where the sum of foreground and background spread is at its minimum [29] [30]. Otsu's method is used to create an initial binary mask that reduces the search space over which the algorithm needs to classify. The foreground, in this case, are higher intensity regions that correspond to high temperatures. Shown in Figure 2.2 (d) is the binary image outputted by Otsu's method and a morphological closing operation.

The sum of intra-class variances that is minimized by $\sigma_W^2 = W_b\sigma_b^2 + W_f\sigma_f^2$ where W_b and W_f are the weights, or proportion of pixels in the background and foreground respectively, and σ_W^2 is the within class variance.

2.1.2.2 Morphological Closing

In the following equations, A represents an image and B is a smaller binary structuring element applied to the image. Erosion which shrinks objects was done by $A \ominus B = \bigcup_{b \in B} A_{-b}$. Dilation which expands ROIs was done by $A \oplus B = \bigcup_{b \in B} A_b$.

In dilation, if B is contained inside A then pixels values are added. In erosion, if B is completely contained inside A then pixels values are retained, otherwise they get deleted. A closing operation is a dilation followed by erosion.

To reduce gaps in each ROI, a morphological closing operation was performed on the mask. A morphological closing operation is operation of dilation of the image followed by erosion [31]. Again, the closing operation result is also shown in Figure 2.2 (d) which acts as a background filter on the original 16 bit image.

2.1.2.3 Distance Transform

The binary mask was passed onto a distance transform routine [32] which was used to find contours that separate and identify the location of each ROI.

The distance transform produces a distance map for each pixel in the image with the distance to the nearest obstacle pixel. The obstacle pixels in binary images are the contours between foreground and background [32]. The peaks become markers for the foreground objects, also known as ROIs.

Equation 2.1 shows the generalized formula for a distance transform where G is an image represented by a grid, P is a set of points defined by a binary image, $I(q)$ in Equation 2.2

indicates membership of P, and D_P is the resulting distance transform image. The distance between two points is represented by $d(p, q)$ and in this research Euclidean distance between two pixels is used. A particular pixel is represented by (x, y), its row and column indexes. The distance to another pixel at (x', y') is calculated by Euclidean distance. Equation 2.3 shows how the formula can be used for images [32].

$$D_P(p) = \min_{q \in G}(d(p, q) + I(q)) \tag{2.1}$$

$$I(q) = \begin{cases} 0 & \text{if } q \in P \\ \text{inf} & \text{if } otherwise \end{cases} \tag{2.2}$$

$$D_f(x, y) = \min_{x', y'}(d(x, x') + d(y, y') + f(x', y')) \tag{2.3}$$

The distance transform is shown in Figure 2.2 (e) where pixels inside the ROI have higher intensities if they are farther away from boundary points. The transform is used for binary image segmentation and identification of the number and center of ROIs in OpenCV [33]. The distance transform used in this research helps encompass and identify ROIs even if they are overlapping or close to each other by defining a clear contour around each ROI. This contour is used to prioritize each ROI by size. Each ROI was sorted from largest to smallest and ROIs with an area below 200 pixels were discarded to ensure successful feature extraction.

2.1.3 Classification of Regions of Interest

Based on the classification of work of Kim et al. [17], four key features (mean, variance, inverse difference moment, and entropy) were determined to be key features needed for classifying fire, smoke, and heat reflections. Using these features, a Bayesian classifier was trained to assign a class to a given ROI. The Naive Bayesian classifier also provides a probability percentage indicating how certain it is that the classification is correct. This percentage is referred to as confidence.

Shown in Figure 2.3 is an overview of the classification process. The possible classes a ROI could fall into were fire, fire reflection, smoke, smoke reflection, hot object, objects, or background. Although these are the labels given to the classes, each class was actually represented by a number.

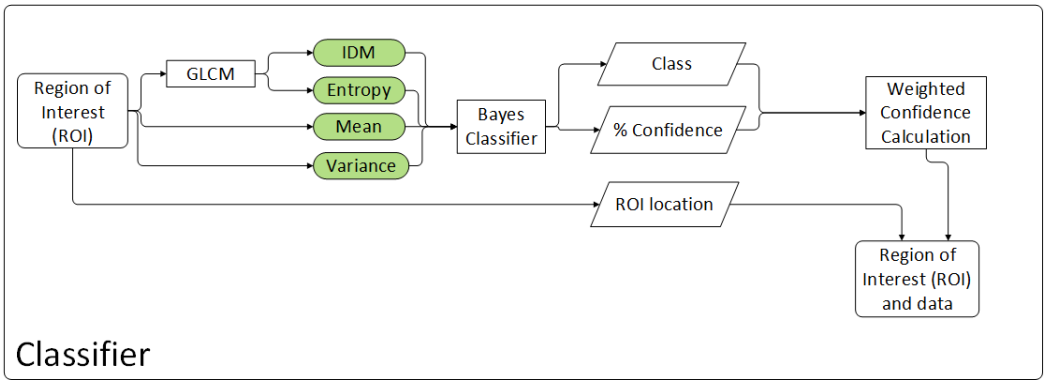

Figure 2.3: Classification of a ROI.

2.1.3.1 First-Order Statistical Features

The two first-order statistical features used in the classifier, mean and variance, were taken using a mask over the original grayscale image.

$$N = \sum_{I:mask(I)\neq 0} 1$$

$$M = \frac{\sum_{I:mask(I)\neq 0} src(I)}{N} \tag{2.4}$$

$$V = \frac{\sum_{I:mask(I)\neq 0} N}{(src(I) - M)^2} \tag{2.5}$$

where I is a pixel in the ROI and N is the number of pixels that are not masked out from a ROI [34]. In this thesis, the mean feature was represented by the variable M and the variance was represented by the variable V.

2.1.3.2 Gray Level Co-Occurrence Matrix (GLCM)

GLCM is a square matrix that acts as a histogram of co-occurring discrete pixel values at a given offset in all cardinal directions. Because of its symmetry, only four cardinal directions are needed to construct it by transposing and adding the matrix to itself [35]. As shown in Figure 2.4, the GLCM was created using a distance of $d = 2$ chosen by Kim et al. [17]. In this case, K is the number of possible discrete values in the image, that is the maximum number allowable by a 14 bit image is $K = (14)^2 - 1$. $P(i, j)$ is a square matrix of size K that keeps

track of the number of times that two pixels are adjacent for different cardinal directions. It is obtained by $P(i,j) = P(i,j,d,0°) + P(i,j,d,45°) + P(i,j,d,90°) + P(i,j,d,135°)$, where the matrix $P(i,j,d,a°)$ keeps track of the number of pixels intensity pairs adjacent to each other at a particular distance and for each angle. The normalized sum of matrices in Equation 2.6 is the GLCM and it is represented by $p(i,j)$.

$$p(i,j) = \frac{P(i,j)}{K} \tag{2.6}$$

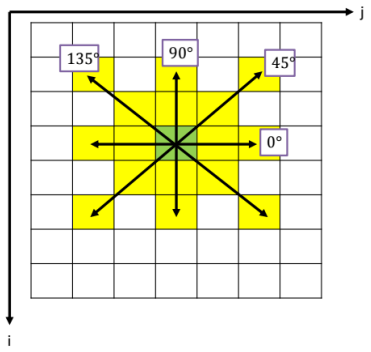

Figure 2.4: A Gray Level Co-occurrence Matrix is created by counting occurrences of adjacent pixels for each pixel in the image.

Entropy and Inverse Difference Moment (IDM) are two texture features that can be extracted from the GLCM. Entropy as shown in Equation 2.7 intuitively shows how much randomness there is in a ROI.

$$E = (-1) * \sum_i \sum_j p(i,j) \log(p(i,j)) \tag{2.7}$$

IDM, shown in Equation 2.8, expresses how much local homogeneity exists in a ROI [35]. In this thesis, the entropy feature was represented by the variable E and the IDM was represented by the variable I.

$$I = \sum_i \sum_j \frac{p(i,j)}{1 + (i-j)^2} \tag{2.8}$$

2.1.3.3 Naive Bayes Classifier

The Naive Bayes Classifier is a supervised learning algorithm that has been successful in classification despite its strong assumptions [17]. Each class and feature is assumed to have

a normal distribution. Therefore, the probability of a feature given a certain class follows a normal distribution as well, as shown in Equation 2.9.

$$F_i \sim N(\mu_{F_i}, \sigma^2_{F_j})$$
$$C_j \sim N(\mu_{C_j}, \sigma^2_{C_j})$$
$$p(F_i|C_j) \sim N(\mu_{F_i|C_j}, \sigma^2_{F_i|C_j})$$

$$p(F_i|C_j) = \frac{1}{\sqrt{2\pi\mu_{F_i|C_j}}} e^{-\frac{1}{2}(F_i-\mu_{F_i|C_j})^\intercal (F_i-\mu_{F_i|C_j})\sigma^{-2}_{F_i|C_j}} \tag{2.9}$$

$$p^k(C_j|F_1F_2...F_q) = \frac{p^k(C_j)p(F_1F_2...F_q|C_j)}{\sum\limits_{C_j}^{k} p^k(C_j)p(F_1F_2...F_q|C_j)} \tag{2.10}$$

$$p^k(F_1F_2...F_q|C_j) = \prod_{i=1}^{q}(F_i|C_j) \tag{2.11}$$

$$c = \underset{C_j}{\operatorname{argmax}}(p(C_j|F_1F_2...F_q)) \tag{2.12}$$

$$p = max(p(C_j|F_1F_2...F_q)) \tag{2.13}$$

Following Bayes rule, the probability of a certain class given a set of features can be given by Equation 2.10. Using a strong conditional independence assumption allows the evidence term to be simplified as a product in Equation 2.11. Using maximum likelihood estimation, the class can be predicted by Equation 2.12. Furthermore, the Naive Bayes classifier outputs a probability, also known as confidence, by Equation 2.13. In this, i keeps track of the total number of q features, j keeps track of the number of classes, and k is the number of candidates or samples. A single sample will have a feature vector F and a class C assigned to it. The predicted class is represented by c and the confidence is represented by p. Confidence is a value between 0 and 1 that indicates how confident the classifier is that its class prediction is a true positive.

2.1.4 Weighted Probability

In this research, a weighted probability measurement is created by combining the predicted class and confidence of classification from the Naive Bayes classifier. The weighted probability is obtained by $WP = w_C * p$ where w_C represents the normalized weight of a

particular class and p represents the confidence obtained from the classifier. The confidence and normalized weight are both measurements between 0 and 1.

Normalized weights, w_C, were assigned to each class by $w_C = \frac{\sum_i^L W(i)}{L}$. $W(i)$ is a priority heuristic assigned to each class and L is the total number of labels. In order to differentiate between classes, arbitrary numbers were assigned to each class at the beginning. These arbitrary numbers were re-organized based on the intuitive priority of leading to fire. The first column in Table 2.1 indicates the priority given to each class, where fire has the largest numeric value and background has the lowest. The classes ranged from 0 to 7 and default of 0 indicates there is nothing to follow in a ROI. w_C in the second column of Table 2.1 is the resulting normalized weight for each class.

Table 2.1: Details on the classes used to calculate normalized weights assigned.

Name	Class by priority, W(i)	Normalized Weight, w_C
FIRE	7	1
FIRE-R	6	0.85
SMOKE	5	0.71
SMOKE-R	4	0.57
HOT OBJ	3	0.42
OBJ	2	0.28
BACKGROUND	1	0.14
NOTHING	0	0

The weighted probability was conceived from the intuition that humans organize objects classes by priority of how likely they are to lead to fire. Intuitively, following smoke would lead to a fire because the existence of fire implies smoke and high temperatures in objects. Hence, to the naked eye, a human would follow smoke until he or she found light reflections from the fire and the fire itself. Although heat reflections cannot be perceived by the naked human eye, firefighters that use hand-held IR cameras can identify hot objects, smoke, and fire.

The weighted probability is then treated as a new probability and a heuristic threshold of $t = 0.5$ was used so that only values above 50% are considered for calculating a heading. This threshold on the weighted probability is calculated for each ROI by Equation 2.14.

$$WP_{ROI} = \begin{cases} 0 & \text{if } WP \leq t \\ WP & \text{if } WP > t \end{cases} \tag{2.14}$$

This final weighted probability calculation is encompassed alongside the ROI raw data and its location inside a data structure. By normalizing the priority and treating it as a weight, both confidence and probability are combined in a weighted probability. The weighted probability

WP balances the need for high confidence and the need to follow a class of high priority.

2.2 Robot Heading

An important realization from looking at the training data for the classifier is that a ROI could take large amounts of space making it difficult to determine a heading. Figure 2.5 (a) contains a hallway filled with smoke which make it difficult for a person to navigate. Figure 2.5 (b) contains the corresponding raw IR image in grayscale.

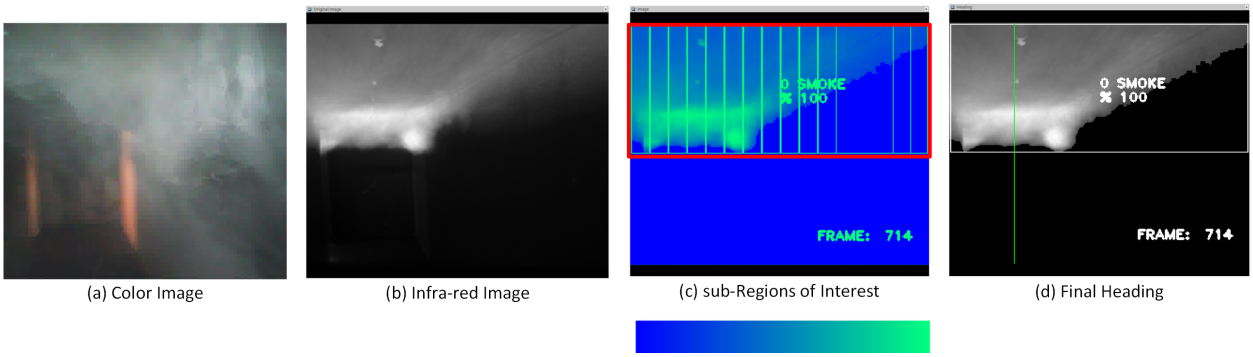

Figure 2.5: (a) Original RGB image (b) FLIRA35 grayscale IR image (c) Partitioning of a ROI (d) Resulting heading angle to guide the robot

The original grayscale image was remapped to a winter colorspace because it is easier to distinguish details in color. The bar below Figure 2.5 (c) indicates that high temperatures are green and low temperatures are blue; this image was used for visualization purposes. In Figure 2.5 (c), the region highlighted in a red box is the original ROI found by auto-thresholding. The label in the middle of the ROI shows that it is classified as *smoke* with 100 percent confidence. This large region encompasses half of the image and following the center of this ROI would lead us to the wall or the corner of the room. Through subdivision of the ROI, the heading is towards the center of the wall at the back of the hallway.

2.2.1 Weighted Probability Heading

This large ROI can be sub-divided into column-wise sub-ROIs and re-classified. In Figure 2.5 (c), the green columns are an overlay made to visualize these partitions. The weighted probability measurement is calculated for each column sub-ROI and partitions that did not have a weighted probability of at least 50% are not shown. The weighted probability calculated for each sub-ROI allows for base of comparison. In Figure 2.5 (d) it is shown that the final heading angle is the sub-ROI with the greatest weighted confidence.

2.2.2 Maximum Temperature Heading

A simple maximum temperature heading was also calculated in this research for comparison with the proposed weighted probability heading. The intensity values in the IR images correspond to temperatures in the FOV; therefore, the intensities were used to determine the maximum temperature heading. The processing included summing the intensities in each column of the image and selecting the column with the maximum intensity. This maximum intensity column corresponded to the heading in that image. To reduce the amount of variation in the heading, a filter was similarly applied to the heading to smooth the result.

2.3 Robots and IR Camera Overview

2.3.1 Tele-operated robot

Real fire data in a room and hallway environment was collected using a mobile robot platform. The robot was equipped with a long wavelength FLIRA35 IR (7–14 micron) camera placed 170 cm above the ground. A human-operator tele-operated the robot by remote control to enter a hallway, move down the hallway, and go into a room containing a fire [17].

2.3.2 Rene

The mobile robotic platform called Rene, shown in Figure 2.6, was used to test the software stack perceiving a fire environment. Rene is an advanced robotic platform capable of manipulation also built by students at Terrestrial Robotics Engineering and Controls (TREC) lab. Rene's upper body is the same as ESCHER; however, its lower body is composed of a PioneerLX mobile platform which communicates with the rest of the system using Robot Operating System (ROS). Rene has been fitted with the two custom seven degree of freedom ROBOTIS Dynamixel Pro arms.

A single long wavelength FLIRA35 IR camera with a temperature range from –40 to +550 C^o and a 9mm lens is used for this research. The small focal length provides a wide Field of View (FOV), 47.92 degrees horizontally and 39 degrees vertically (48x39). The raw input for the algorithm is a 14 bit unsigned grayscale image embedded in 16 bits. The camera transmits 320x256 grayscale images at a rate of 60 Hz, weights 0.2 Kg, and consumes 3.5 W [36]. An Ethernet camera driver called camera_aravis was used to interface the camera through ROS.

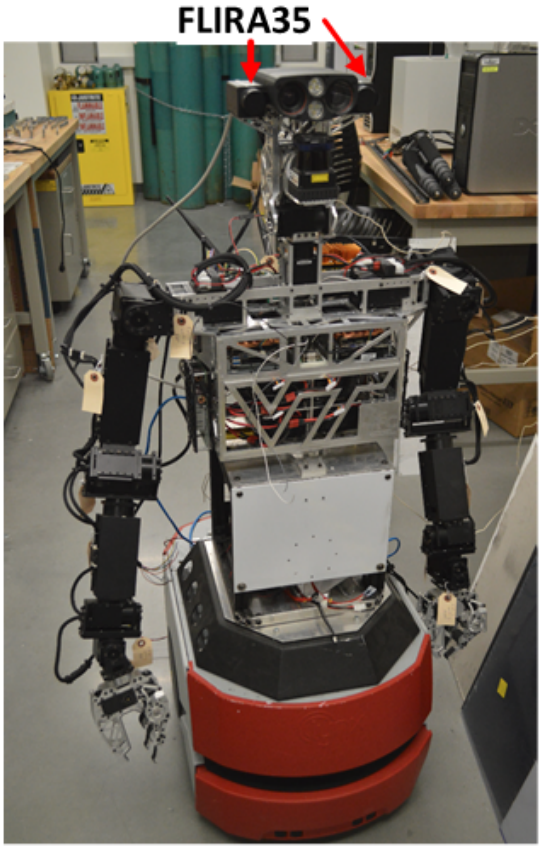

Figure 2.6: Rene was build at TREC for the SAFFiR project.

Chapter 3

Fire Tests

3.1 Room-Hallway Large-Scale Tests

In order to collect real fire environment data, a human-operator tele-operated a mobile robot platform to enter a hallway, move down the hallway, and go into a room containing a fire [17]. The environment itself is shown in Figure 3.1. Seven different real fire environment data were utilized in this thesis in order to extract features to train the classification algorithm and provide data to test the heading algorithm.

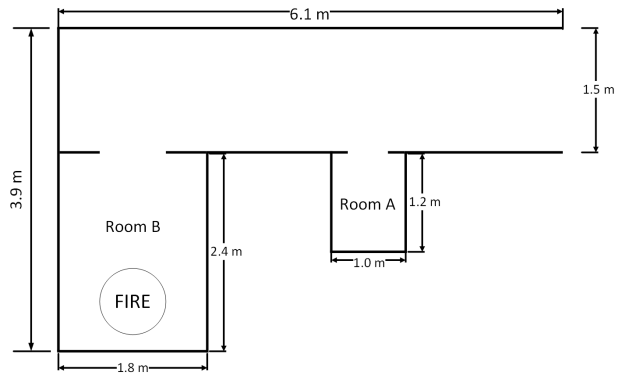

Figure 3.1: Lab room, hallway, and fires used to collect variations of fire-environment data.

3.1.1 Data from seven different fire environments

Variation in the data set ensures that training a classification algorithm, such as the Naive Bayesian classifier, would not over fit. A total of ten raw-data videos containing different fire and smoke conditions were collected by Kim et al [17]. Seven videos stored in .DAT

Table 3.1: Fire test data details.

Name	Fuel Type	Total Time	Temperature	Smoke Visibility
2_HG_Latex_Below_Smoke	Latex	177 (s)	Low	Low
3_HG_Latex_Below_Smoke	Latex	179 (s)	Low	Low
4_HG_Latex_Lights_Switched	Latex	169 (s)	Low	Low
7_HG_Wood	Wood	112 (s)	Low	Medium
11_HG_Latex_Late	Latex	69 (s)	Medium	Medium
13_HG_Propane_25kW	Propane	80 (s)	Medium	High
14_HG_Propane_75kW	Propane	72 (s)	High	High

formats were translated to Rosbag files and used for feature extraction and training the classification algorithm in C++ and Robot Operating System (ROS). Table 3.1 contains fire and smoke conditions used in the experiments. The temperatures range from at least 0 C^o to 550 C^o in the different tests. Smoke visibility is the distance a person can see an object through the smoke, with low visibility resulting from more dense smoke. Minimum visibility in the tests ranged from 0 m to over 6 m. Latex fires, which correspond to the first three videos, are known to have low smoke visibility and low-temperature fires. The file titled *4_HG_Latex_Lights_Switched* in particular was taken with the lights switched off. The file titles *7_HG_Wood* was a wood crib fire started using acetone. Wood fires produced low to medium gas temperatures with medium density smoke. Finally, the last two videos are two propane fires resulted in medium and high gas temperatures. Both propane fires produced low amounts of smoke resulting in high visibility.

3.2 Feature Extraction

As discussed in Chapter 2, each image was segmented in order to find ROIs in the grayscale image. The ROIs extracted for training were not sub-divided. Instead, these ROIs were taken after the first classification step, shown in Figure 2.1, and they were identified by order of size; larger ROIs would be processed first. In addition, each ROI can be described by a rectangular polygon and a mask, and for each ROI the four key features were extracted in the same manner as described in Equation 2.7, Equation 2.8, Equation 2.4, and Equation 2.5. The ROI identifier and its four features were saved to .csv files for each training video. In order to correctly identify where the ROI came from, the time of the frame and the mean of the overall grayscale image were also saved. With a total of 9,110 samples, the grayscale images were correlated to RGB videos taken at the fire lab for each test and labeled as belonging to one of 7 classes. Class 8 was reserved in order to erase a sample that was difficult to classify or meaningless, for example, when the camera shutter would black out an image. Table 3.2 includes the number of samples labeled for each class number and the name field contains the translation of the numeric identifier to the name of the class. The class

ID in Table 3.2 were originally used to distinguish between classes and thus were arbitrarily assigned to train the classifier. Something important to note about the video data is that most of the training images contained a single ROI. This is because smoke covers half of the image and this region is encapsulated into one region.

Table 3.2: Different object classes and the number of ROI in each class for the training set.

Name	Original Class ID	Number of ROIs
HOT OBJECT	7	82
BACKGROUND	6	1687
FIRE-REFLECTION	5	338
FIRE	4	1517
SMOKE	3	4844
SMOKE-REFLECTION	2	393
OBJECT	1	249

3.3 Laboratory Tests

Because the training data was taken at the same environment, an additional four fire tests were set up in order to test for over-fitting and the successful detection of classes. Figure 3.2 contains details on how the each of the four tests were set up. The tests consisted of placing one of the burning materials on a metal pan below a fume hood. The burning materials are not visible but they are held up by the black metal frame in Figure 3.2. The two burning materials chosen were latex foam and a wet paper towel covered over by white paper. Furthermore, a ceramic backboard was used as a variable to test the effect of the environment in the original training data. The conditions of each test are summarized in Table 3.3. Two camera video feeds, RGB and IR, recorded each fire test for comparison. High density and low density smoke similar to that in the training data were used to validate that the classifier could detect fire, smoke, and heat-reflections successfully. The backboard was used to test whether or not the algorithm was over-fitting to the environment of the training data.

Table 3.3: Description of the laboratory fire tests.

Name	Total Time	Smoke Density
Wet paper with backboard	291 (s)	Medium
Wet paper no backboard	259 (s)	Medium
Latex fire with backboard	108 (s)	High
Latex fire no backboard	91 (s)	High

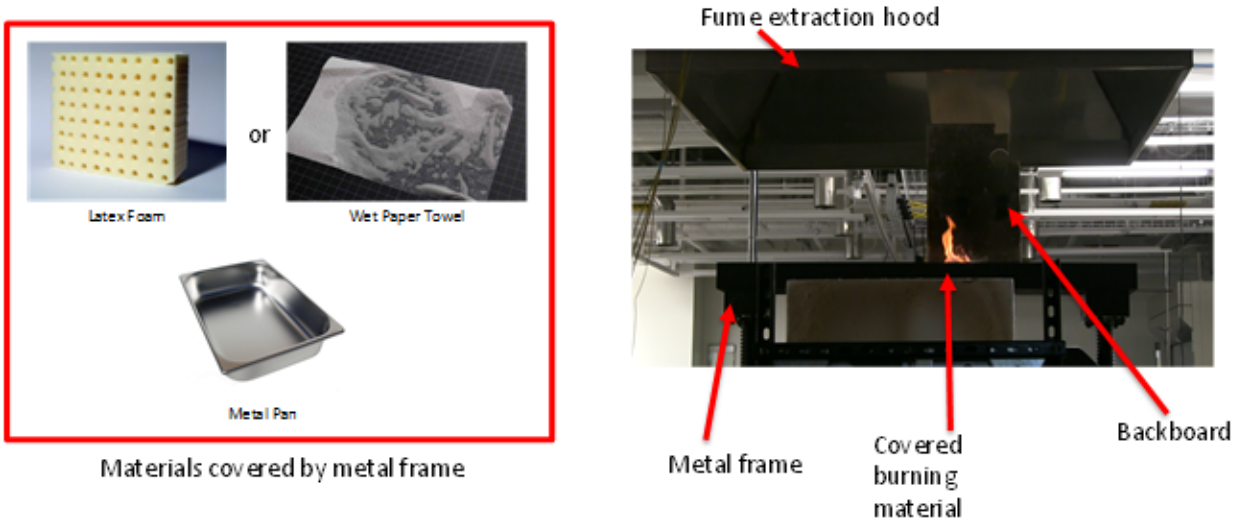

Figure 3.2: Indoor laboratory fires tests to validate classification in a different fire-environment.

Chapter 4

Classification Training and Validation

4.1 Classifier Model

The Naive Bayes classifier forms an model for each class using Gaussian parameters. The classifier uses four features (entropy, IDM, variance, and mean) distributions to determine which class a new sample would belong to. The following three tables contain the parameters that describe the model for the Naive Bayes classifier produced based on the training data. Table 4.1 includes the Gaussian averages μ values for each feature and for each class. Table 4.2 contains the Gaussian standard deviations σ for each feature and for each class. Finally, Table 4.3 contains the *a priori* beliefs that a certain class will show up for each class.

Table 4.1: Mean of the distributions μ for each feature.

CLASS	MEAN	VARIANCE	ENTROPY	IDM
Smoke-ref	3057.6	51289.1	1.9693	0.6363
Smoke	4085.5	263234.5	1.6796	0.8438
Fire	12278.5	5070916.3	2.0873	0.6272
Fire-ref	10004.1	1563246.1	1.8673	0.7168
Background	2663.5	9052.7	1.7152	0.7850
Hot Object	2875.9	23332.0	1.8780	0.6127
Object	2906.4	37910.4	1.8147	0.7302

4.1.1 Classifier Model Distributions

The Gaussian distributions are provided in Figure 4.1 for four key classes as well as the the four features (entropy, mean, variance, and IDM). The four key classes are fire, fire-reflections, smoke, and smoke-reflections. The first feature, mean, represents average

Table 4.2: The standard deviation σ for each feature.

CLASS	MEAN	VARIANCE	ENTROPY	IDM
Smoke-ref	437.2	201215.7	0.2056	0.1187
Smoke	776.0	275434.9	0.1858	0.0711
Fire	1571.9	1533091.5	0.1438	0.0838
Fire-ref	1476.3	2565756.4	0.2565	0.1241
Background	479.5	38168.2	0.1591	0.0379
Hot Object	336.7	119713.9	0.2693	0.1091
Object	282.3	36656.6	0.2335	0.0599

Table 4.3: The prior probabilities for each class.

ID & CLASS	PROBABILITY
1 Smoke-ref	0.0273
2 Smoke	0.0431
3 Fire	0.5316
4 Fire-ref	0.1664
5 Background	0.0370
6 Hot Object	0.1853
7 Object	0.0089

temperature intensities for each class. Fire has the highest average temperature whereas fire-reflections, hot objects near fire or containing fire elements, have the second highest temperature. Smoke-reflections are hot objects that exist in the presence of smoke such as heated walls. In a similar manner, smoke has higher temperatures than smoke-reflections. In terms of variance of pixel values, both fire and fire-reflections have a wider range than both smoke and smoke-reflections. The texture features also show key differences to aid the classification of the four classes. Entropy represents the randomness in the ROI and IDM represents the local homogeneity. Because heat-reflections include objects like the heated wall, both are similar in texture and in between the values for smoke and fire. Fire has more entropy than smoke while smoke has a higher IDM than fire.

4.2 Evaluation

4.2.1 Confusion Matrix

The performance of the classification of fire, smoke, and heat reflections is analyzed using precision, recall, f-measure, and g-measure. These four classes are emphasized because they drive the localization algorithm. The measurements are extracted from a confusion matrix

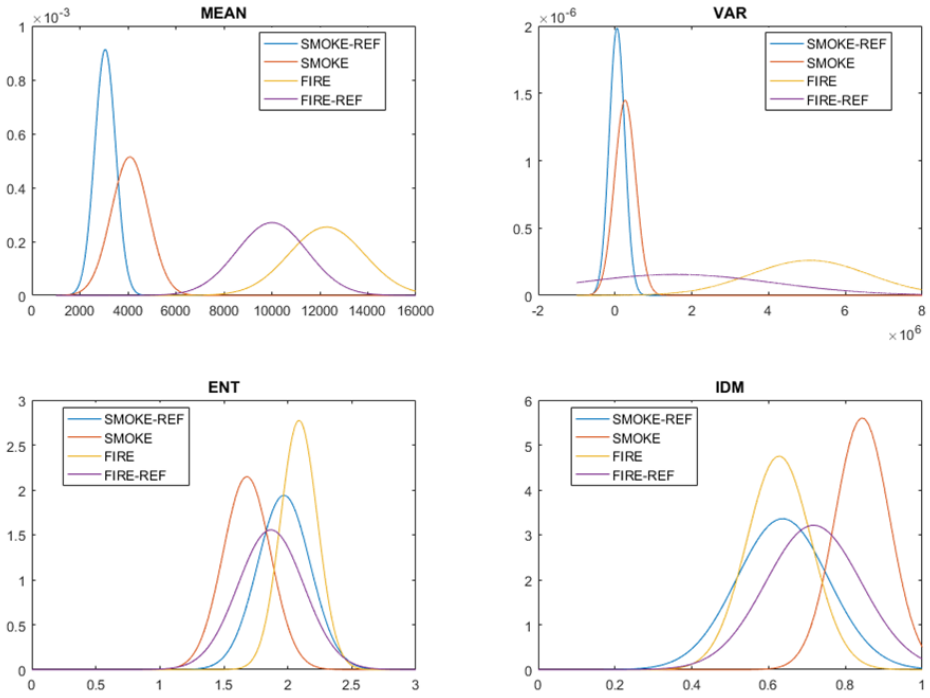

Figure 4.1: Gaussian distribution models for the four key classes.

as shown in Figure 4.2.

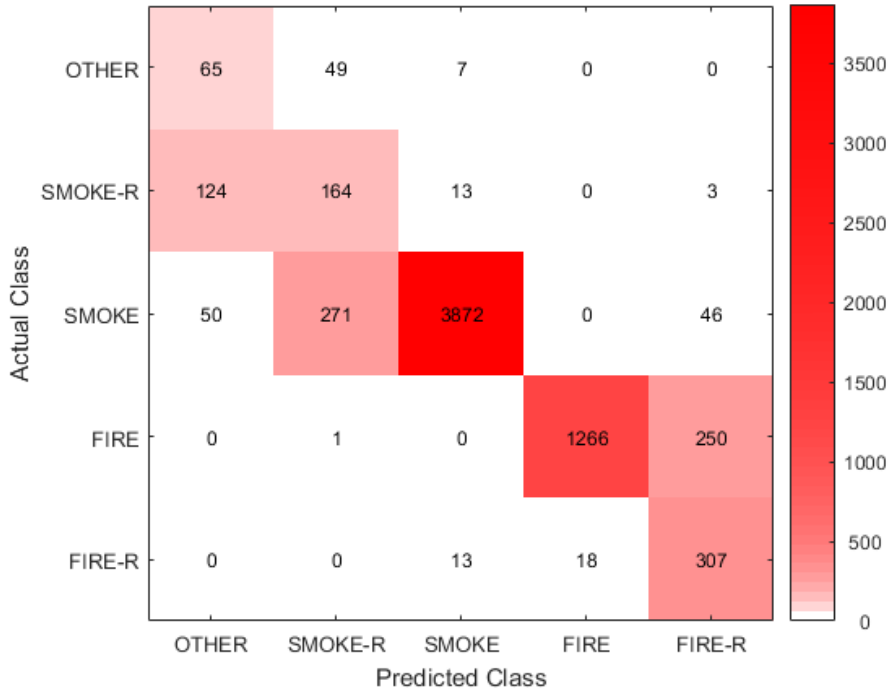

Figure 4.2: A confusion matrix of the classified fire environments.

4.2.2 Precision, Recall, F-score, G-score

Precision is a measure of the proportion of correctly classified objects, or true positives, out of the total number of objects predicted to belong to that class. The total number includes incorrect predictions, or false positives, that should have belonged to other classes. It is a way to measure how often the predicted results are correct.

Recall is a measure of the proportion of correctly classified objects out of the total number of objects that actually belong to that class. Some of the objects that belong to the class of interest may have been misclassified as other classes; hence the sum of false positives and true positives represents the actual number of objects belonging to a particular class.

The precision and recall were calculated by Equation 4.1 and Equation 4.2 using the data from the confusion matrix.

$$precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$recall = \frac{TP}{TP + FN} \tag{4.2}$$

Equation 4.3 provides a weighted combination of precision and recall (F-measure) while Equation 4.4 includes the geometric mean of precision and recall (G-measure).

$$F = 2 * \frac{precision * recall}{precision + recall} \tag{4.3}$$

$$G = \sqrt{precision * recall} \tag{4.4}$$

Table 4.4 includes recall, precision, F-measure, and G-measure for the four classes. The evaluating measurements and confusion matrix show that smoke-reflection is the least reliable classification model because it is often classified as smoke or a hot object. Fire reflections are often misclassified as fire but has higher recall, precision, f-measure and g-measure than smoke-reflections. However, smoke and fire have strong values across all four measurements.

Table 4.4: Evaluation of fire environment classification.

class	recall	precision	F-measure	G-measure
smoke-r	41.7	29.2	34.4	34.9
smoke	87.2	98.1	92.3	92.5
fire	83.5	98.6	90.4	90.7
fire-r	90.8	50.7	65.0	67.8

A way to improve the algorithm would be to obtain more samples. As shown in the confusion matrix in Figure 4.2, smoke-reflection has 485 labeled samples and fire-reflection has 606 samples. By comparison, smoke has 3905 labeled samples and fire has 1284 samples. Fire-reflections are often misclassified as fire and have a precision of 50%. Smoke-reflections are often misclassified as smoke and have a precision of 29%. For smoke-reflections in particular, there are more samples misclassified as smoke than the number of true positives. By comparison Kim et al. obtained more samples of both heat reflections by extracting out more ROIs [17].

4.3 Laboratory Test Validation

The four fire tests were designed to test how well the classifier could classify fire-related elements in a different environment. Previous empirical tests with incense smoke demonstrated that smoke was a difficult element to detect in a laboratory setting. As mentioned

in Chapter 3, wet paper fire produced medium density smoke whereas latex fire generates higher density smoke.

The following image, Figure 4.3, contains plots from each laboratory test. These plots were created by recording the class and weighted probability of the optimal sub-ROI chosen by the algorithm in each frame. A threshold of 0.5 ensures that only predictions with weighted probabilities above the threshold can be used for the localization algorithm. Tests where a background placed behind the fire is present improve the variance and number of classifications. Particularly for Class 3, smoke, having denser smoke from latex foam improves the variance of smoke classifications. The weighted probabilities varied from the lowest allowed value of 0.5 to the highest 1.0.

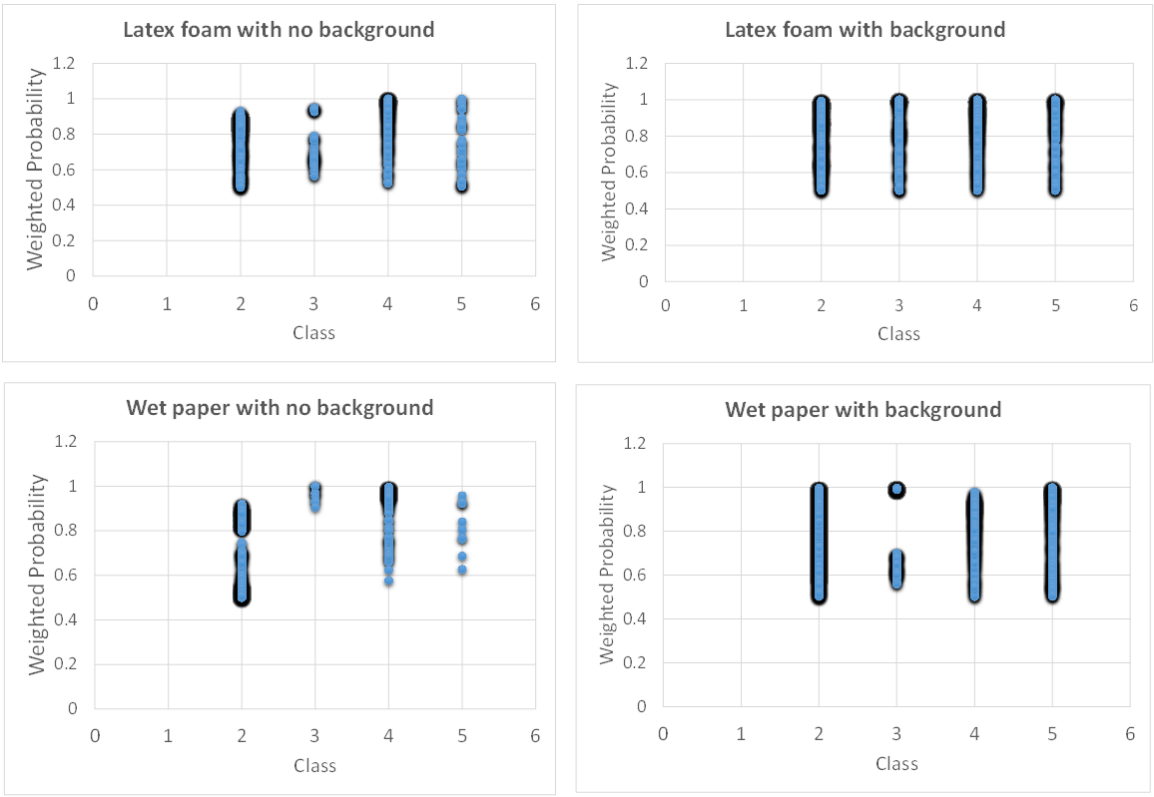

Figure 4.3: Weighted probability vs. class in each of the laboratory tests.

Figure 4.4, includes plots of classes and weighted probability perceived for each laboratory experiment. These plots were generated by recording the class of the optimal sub-ROI chosen by the algorithm and its respective weighted probability for each frame. Each fire test described in Chapter 3 had a burning material that eventually burned out. The plots successfully shows the progression from fire to smoke as the fire dies down except in the case with wet paper with no background.

The total number of detections and variance in the plots and Table 4.4 show that the drywall

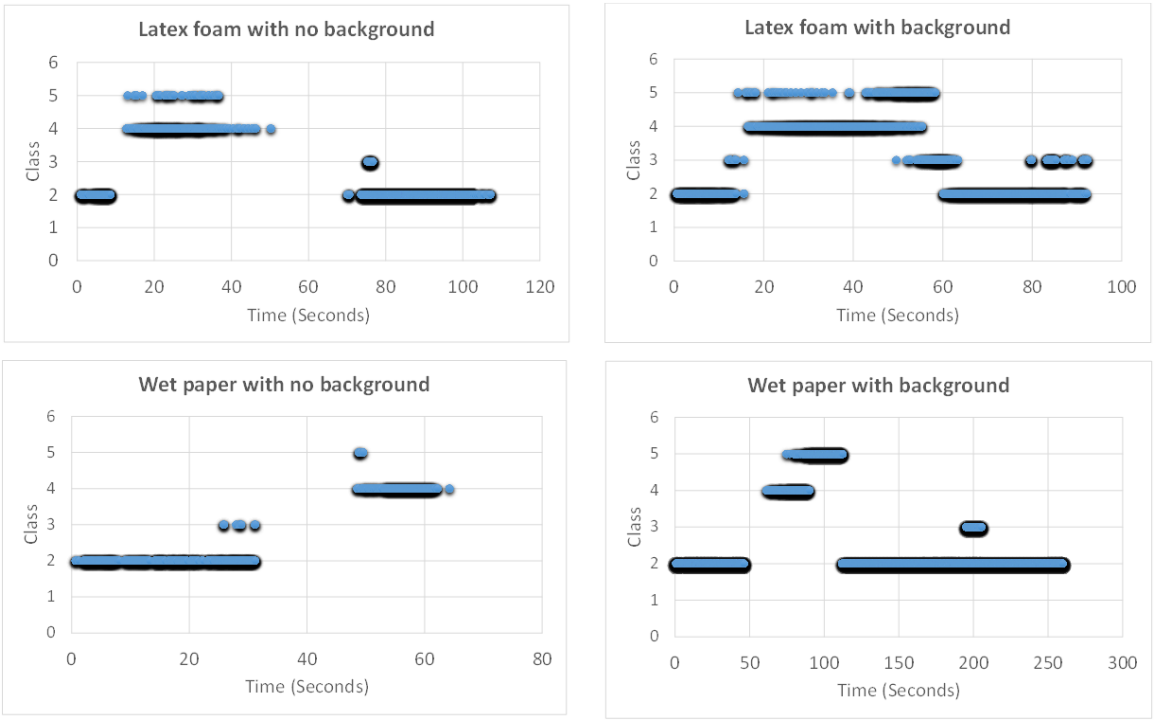

Figure 4.4: Classifications over time of sub-ROIs with the highest WP chosen to calculate heading.

Table 4.5: Number of classifications of chosen heading ROI in four laboratory test settings.

	Wet paper		Latex Fire	
background present?	no	yes	no	yes
smoke-r	1891	15177	6954	2926
smoke	19	546	83	516
fire	581	829	604	1438
fire-r	10	1419	50	428

background improves the detections of smoke and fire. Furthermore, as the fire dies down, the probability of correct classification decreases and as the smoke diminishes the probability also decreases. These empirical observation results are based on comparisons to the RGB camera video feed. Images from each tests are shown in Figure 4.5, Figure 4.6, Figure 4.7, and Figure 4.8. In the wet paper with no background case in particular, the image froze after the fire went out and the smoke was not perceived.

Laboratory tests show that when perceiving smoke, it is necessary to have a heated background to successfully extract features and encompass smoke in a ROI. This is an important discovery of a limitation of this algorithm because it shows that the algorithm would not work in an open environment with low amounts of smoke such as incense smoke.

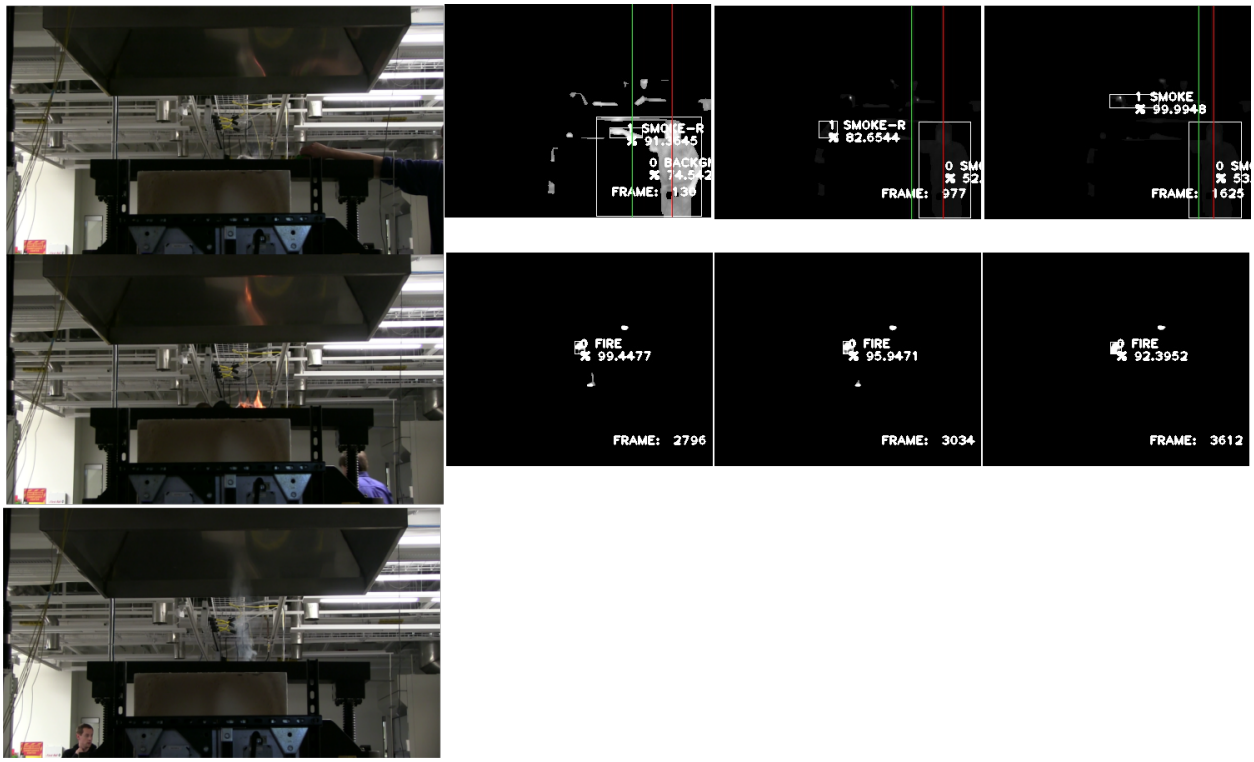

Figure 4.5: Wet paper with no background case.

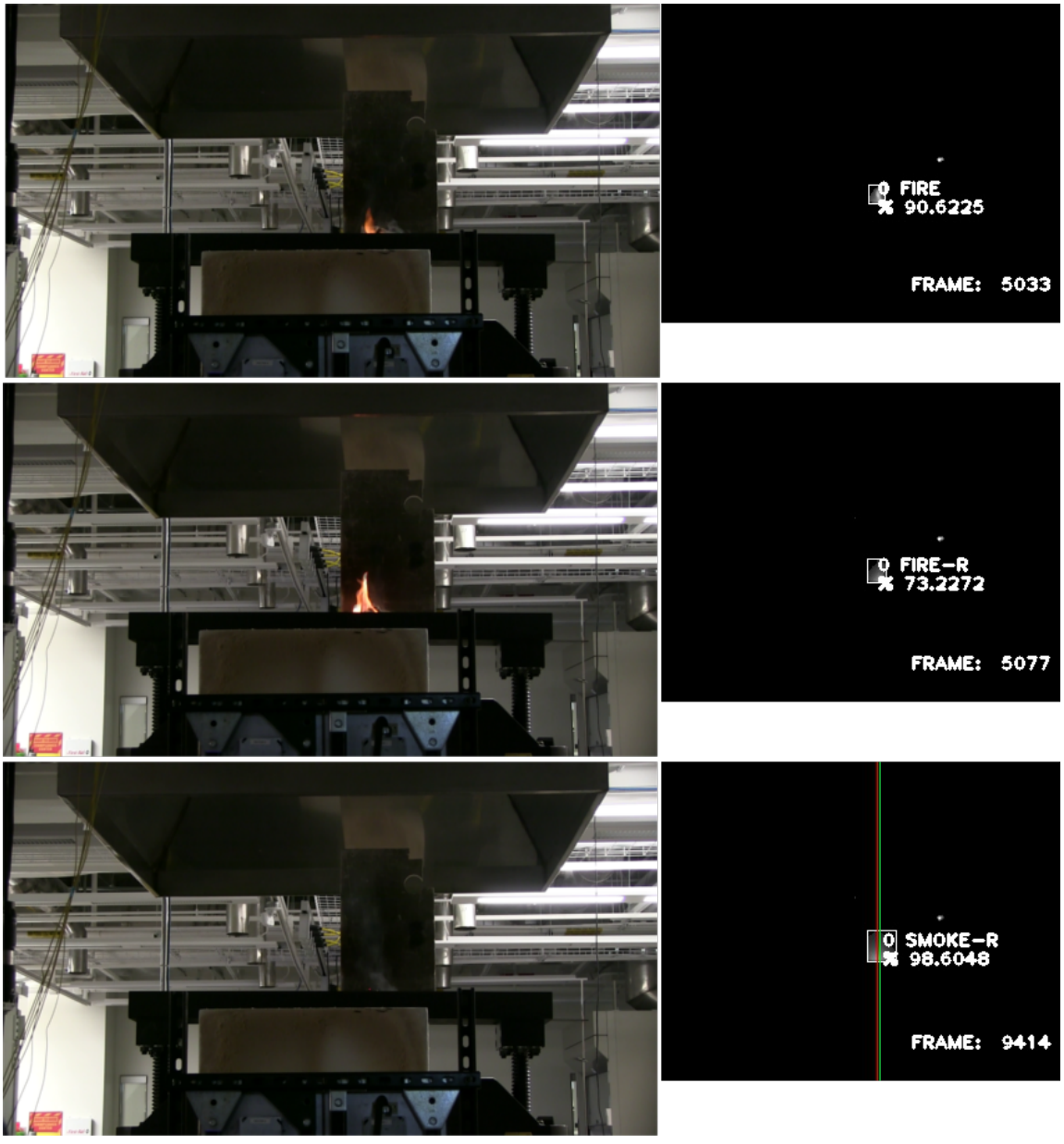

Figure 4.6: Wet paper with background case.

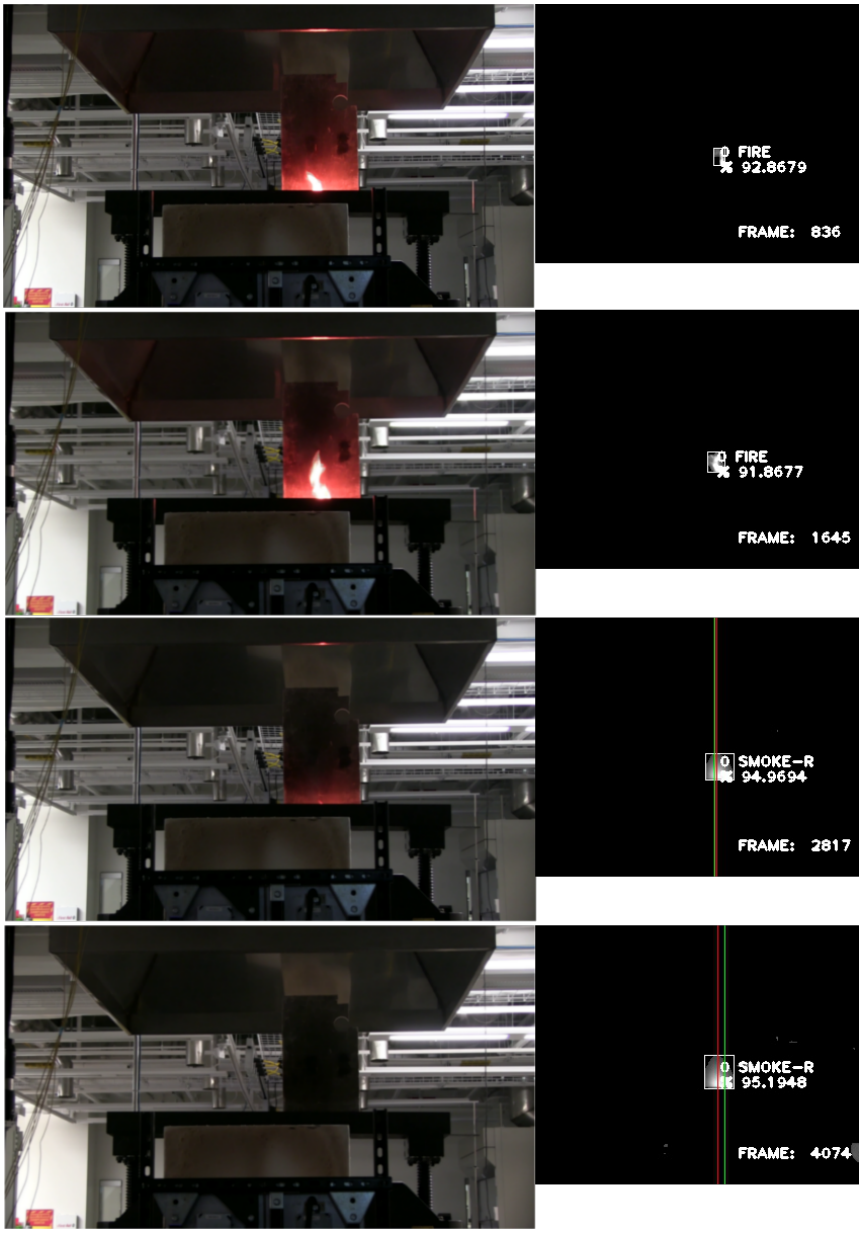

Figure 4.7: Latex with background case.

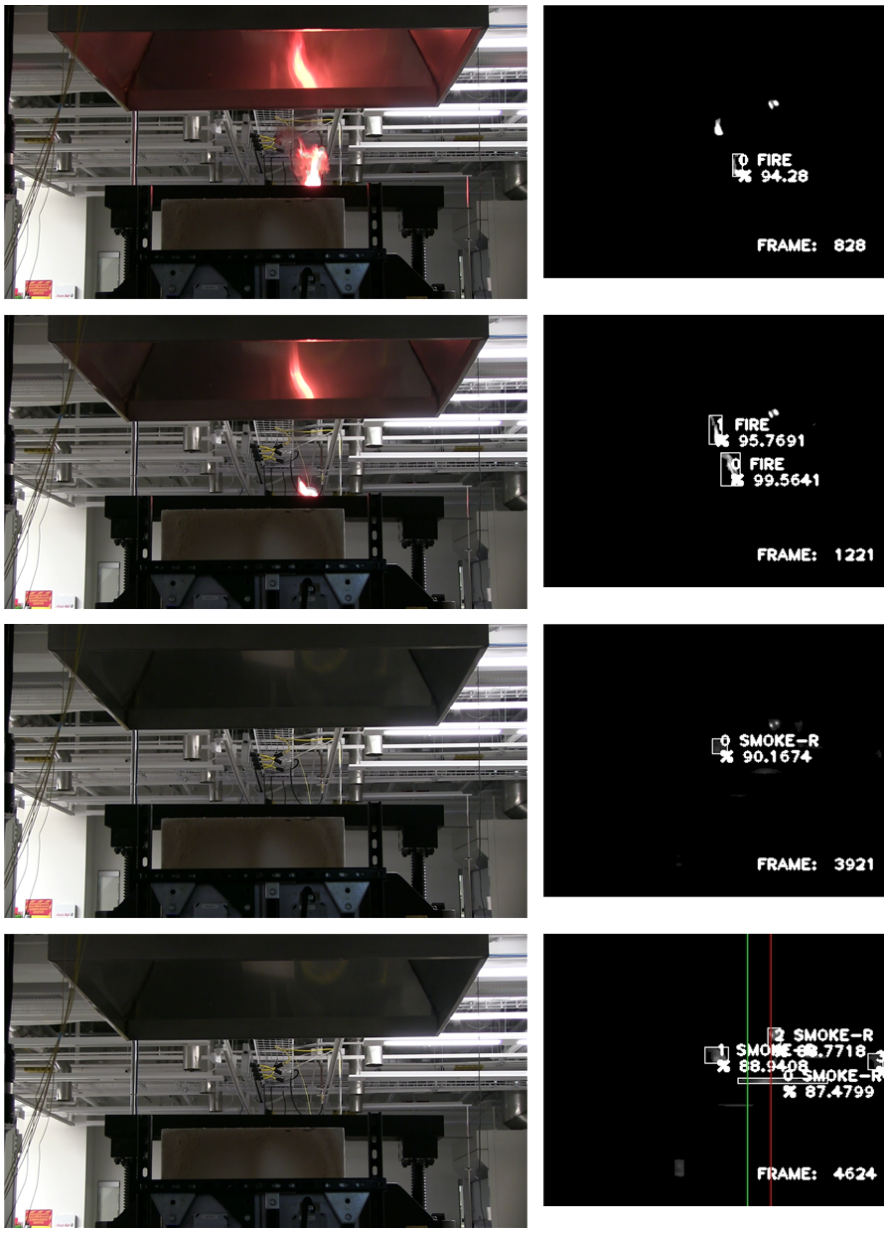

Figure 4.8: Latex with no background case.

Chapter 5

Robot Heading Using Hallway Test Data

5.1 Overview

The heading from the weighted probability algorithm was evaluated using IR data previously collected by [17] using a mobile robot in the room-hallway fire test facility shown in Figure 3.1. The videos were played as though a robot was perceiving them to determine a heading angle with the weighted confidence algorithm. These headings were compared with a maximum temperature intensity algorithm and a perceived tele-operated heading.

5.1.1 Yaw Heading Angles from Maximum Temperature and Estimates

The maximum temperature heading was obtained by following the highest column-wise sum of pixels on the IR image. The highest sum of pixels indicates the highest temperature. The angle calculated for the temperature heading were smoothed out by a running average of five temporal images. As for the manually estimated heading, angles were determined from rectified IR images with a maximum and minimum of positive/negative 20 degrees. By convention, the center of the image signifies a yaw of 0 degrees; turning left corresponds to a positive yaw angle and turning right corresponds to a negative yaw angle as shown in Figure 5.1.

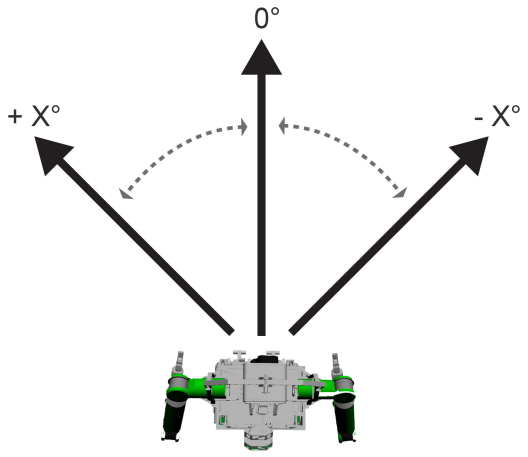

Figure 5.1: Robot yaw allows for a heading angle to control the robot. A positive angle corresponds to turning left and a negative angle corresponds to turning right.

5.2 Results

Predictions were conducted using the training data videos which included wood crib, latex foam, and propane fires in the fire lab. Details of each video are discussed in Chapter 3 and Table 3.1. Each of the following images contains a time lapse of fire test, taken by an IR camera and the RGB camera of the test, as well as a plot comparing the three headings: Figure 5.2 Figure 5.3 Figure 5.4 Figure 5.5 Figure 5.6 Figure 5.7 Figure 5.8. The plots have been cropped to analyze only the hallway data. The headings obtained by weighted confidence are shown in green, the headings obtained by maximum temperature are shown in red, and the manually acquired estimated headings are shown in black. Estimated headings were determined by manual review provided and these estimated headings were used as ground truth. All videos ran in real-time and the average computation time of the weighted probability algorithm, from acquiring the image to the output of the heading, was 9.51 ms per image.

Each of the following images contains a time-lapse of the hallway tests and the resulting headings. The images also contain the perceived class and weighted probability chosen by the localization algorithm for the heading of the robot. To the bottom right is a frame reference counter that can be used to match the timing on the plot. An important note is that the timelapse screen shots were taken with a threshold of 50% on the weighted confidence. This threshold was not applied on the latex late (11 HG) test in order to obtain a heading. In all videos, the estimated heading follows door of the room with fire. Although the original videos do not guarantee that starting points for the tele-operated robot, the

robot followed the same pattern of movement: a left turn at the beginning to face the center of the hallway, a traversal down the hall, and a final left turn to face the fire.

5.2.1 Yaw Heading Angles From (02 HG) Latex Fire

The weighted probability heading is seen in Figure 5.2 has the same trend as the estimated heading but consistently produces a lower heading angle. Compared with the estimated heading during the 100–149 second period, the weighted probability had an average degree difference of 10.8 degrees while the temperature heading had an average degree difference of 12.9 degrees. During the 149–183 second period, temperature heading performed better with an average difference of 1.3 degrees compared with the weighted probability which differed by 7.6 degrees. During the last stage (183–205 seconds), the temperature heading differs by 30.6 degrees but weighted probability differs by 12.7 degrees. The jump in angle from the temperature heading at 150 and 183 seconds is caused by the hot gases accumulated at the corner of the end of the hallway. As the robot got closer at 150 seconds, the temperature heading angle followed the door containing the fire because its temperature was high. At 183 seconds, the door was not on the field of view so the gases at a corner of a room became the hottest region to follow. Temperature heading is prone to false headings from hot smoke accumulations which would not lead to the fire.

Figure 5.2 (b), (c), (d), and (e) includes time-lapses for the latex fire (02 HG) run, the images on top were taken from a color camera and the images on the bottom are annotated IR images input into the algorithm. Figure 5.2 (a) both headings agree on a left positive angle to enter the hallway. In Figure 5.2 (f) both headings agree on the location of the fire. While temperature heading can accurately output headings that lead to the fire, it is more erratic than the weighted probability heading. In this case, the weighted probability outperforms the temperature heading by providing a more consistent left yaw.

5.2.2 Yaw Heading Angles From (03 HG) Latex Fire

Figure 5.3 contains headings for the 03 HG Latex Fire tested in the hallway-room environment which produced dense smoke and low gas temperatures. This test is similar to 02 HG and can be considered a repetition. In the Latex Fire (03 HG) plot (90–140 seconds) both headings do not match well with the estimated heading as temperature deviates on average by 12.9 degrees and weighted probability heading deviates by 10.4 degrees. Between 140–210 seconds, the temperature heading agrees better with the estimated heading with a 0.32 degree deviation compared to a 10.0 degree deviation from weighted probability. At the end (210–213 seconds) of the hallway, the temperature heading points to the hot gases at the right end of the hallway by comparison the weighted probability heading keeps the same left yaw. Again, the jump in temperature heading at 140 seconds is due to the fact that the robot is close enough to the door for this region to become the hottest object in

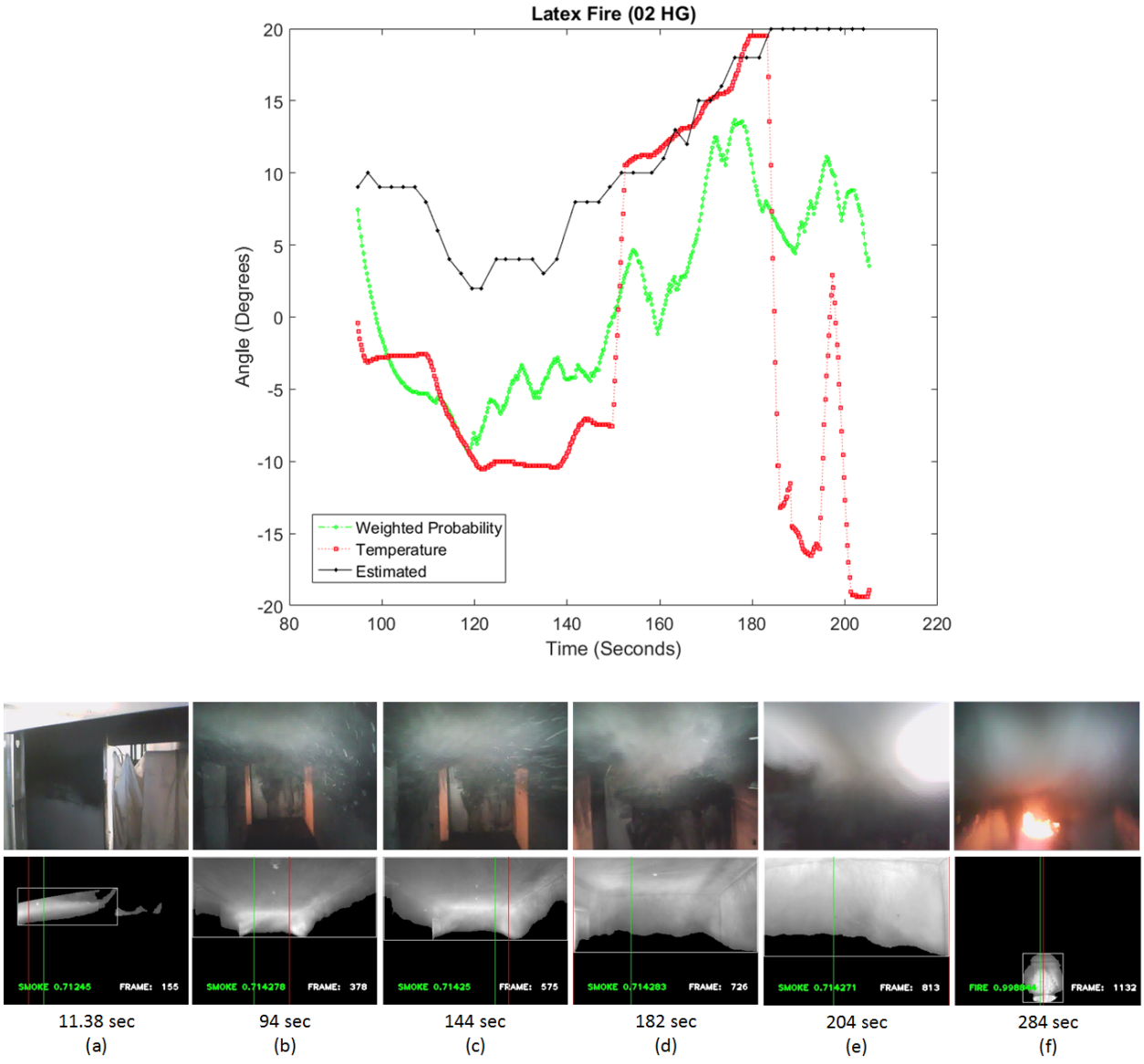

Figure 5.2: Latex fire (02 HG) time lapse and headings comparison.

the FOV. The second jump at 113 seconds for temperature heading occurs because the door is no longer visible at the end of the hallway. Generally, the weighted probability heading looks similar to the estimated optimal heading but shifted by around 10 degrees. Due to the erratic behavior of the temperature heading, the weighted probability heading outperforms the temperature heading.

At 145 seconds in Figure 5.3, the temperature heading switched towards the correct door as shown in the 159 second image. At the end of the hallway, as shown at 213 seconds in Figure 5.3, both the temperature and weighted confidence have a preference for the left side but they are not shown because the weighted confidence heading fell below the 50% threshold. Additional images in Figure 5.3 show the progression of the robot entering the hallway and also correctly classifying fire.

5.2.3 Yaw Heading Angles From (04 HG) Latex Fire With Lights Switched Off

Figure 5.4 contains the Latex Fire (04 HG) plot from 100–147 seconds, the weighted probability performs better (2.9 degree deviation) than temperature heading (12.6 degree deviation). From 147–199 seconds, the temperature heading does better (0.16 degree deviation) than the weighted probability heading (11.5 degree deviation). Once again, the robot is close enough to the room with fire for the temperature heading to detect it as the hottest place on the FOV. In a similar manner as the previous examples, the temperature heading switches at 199–206 seconds because of the hot gases on the right side of the hallway. From 190–210, the temperature heading has changed from positive 20 degrees to negative 20 degrees, a 40 degree change, compared to the weighted probability with a 5 degree change. The weighted probability is less prone to high deviations which would guide a robot smoothly and would keep the room with fire on the FOV. In Figure 5.4 at 206 seconds, the RGB camera shows dense smoke but the IR image clearly shows the end of the hallway.

5.2.4 Yaw Heading Angles From (07 HG) Wood Crib Fire

Figure 5.5 contains the wood fire (07 HG) plot which has a similar trend to the (04 HG) plot. From 69–97 seconds the weighted probability does better (1.7 degree deviation) than temperature heading (14.3 degree deviation) at the beginning. Then, at 97- 128 seconds the temperature heading performs better (0.62 degree deviation) than weighted probability (7.9 degree difference). As shown Figure 5.5 at 63 and 112 seconds, the switch from the temperature heading angle happens because, again, the robot is close enough to the room with fire. The temperature heading tracks the room with fire as long as it remains in its field of view. However, as the robot faces the wall at the end (128–145 seconds) the weighted probability heading performs better (14.4 degree deviation) than temperature (30 degrees deviation). Even when faced with the ambiguity of the wall with smoke, the weighted

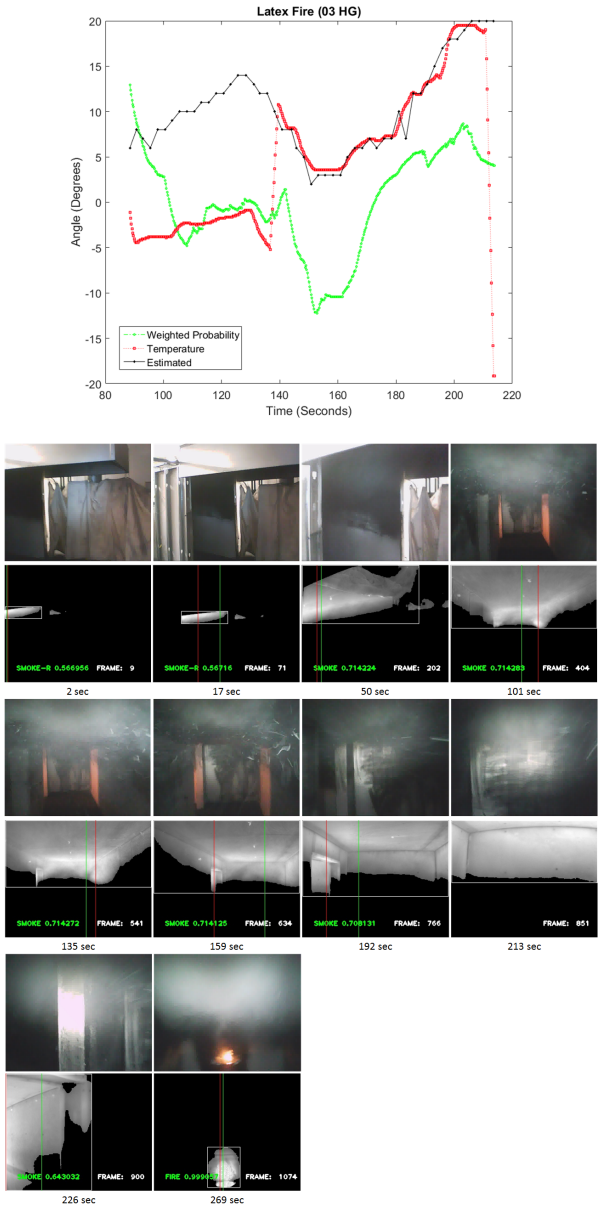

Figure 5.3: Latex fire (03 HG) time lapse and headings comparison.

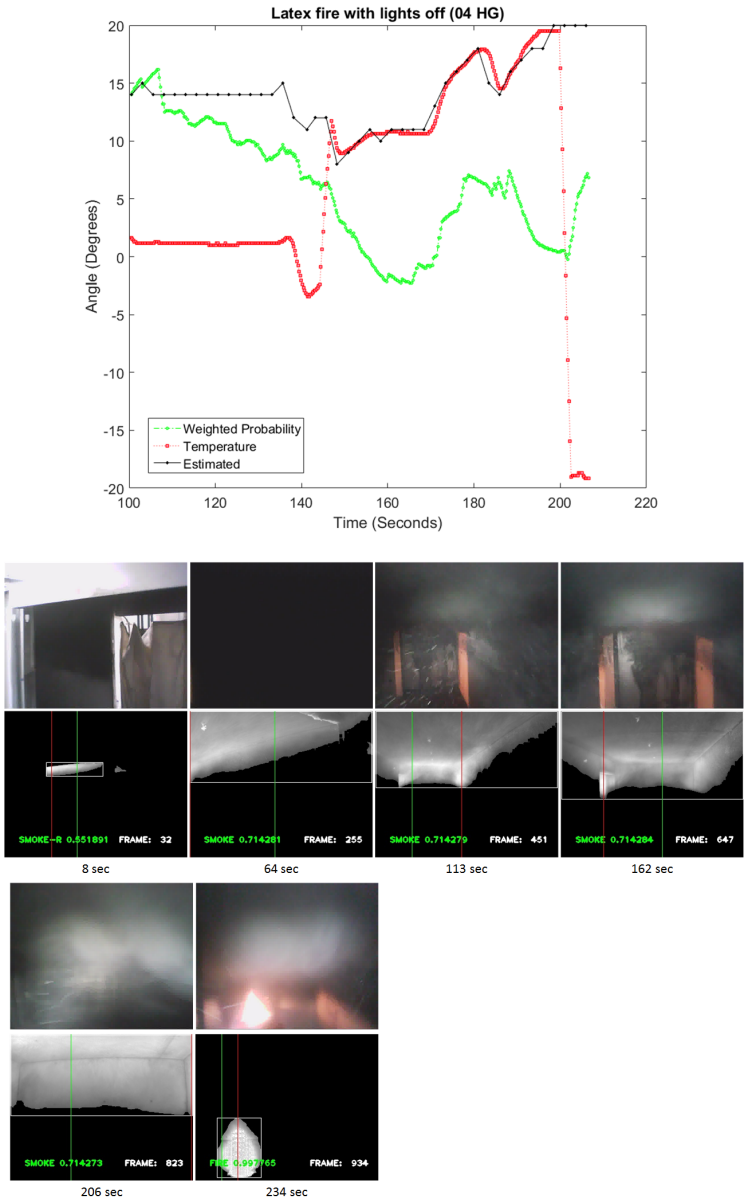

Figure 5.4: Latex fire with lights switched (04 HG) time lapse and headings comparison.

probability heading outputs a positive degree angle throughout the room. The weighted probability heading outperforms the temperature heading because it would lead the robot towards the left where the room with fire resides.

5.2.5 Yaw Heading Angles From (11 HG) Latex Late Fire

The latex late fire (11 HG) data in Figure 5.6 had to be taken with no threshold because weighted probabilities were low for the hallway event. This test in particular had higher density smoke but lower temperatures compared to propane gas at 25kW (13_HG). The difference in temperature made it difficult to detect classes by the weighted probability algorithm. Results show that in this test, the weighted probability did not perform as well as the temperature heading and that the weighted probability heading had an almost constant positive 4 degree heading. This test shows that the algorithm would not work in cases of low temperatures such as at the beginning of a fire event. Because other work explores using RGB cameras for detection, sensor fusion could be an option to explore in order to obtain classification information. Images in Figure 5.6 had been set up to show a heading only when a weighted temperature heading exists, however, the plot shows that temperature heading outperformed the weighted probability heading. This is because the room with fire was the hottest region in the FOV and the temperature of smoke did not reach temperatures that could compete with the temperatures at the door. A hybrid approach could benefit the weighted probability algorithm in cases where classifications are not found.

5.2.6 Yaw Heading Angles From (13 HG) Propane Gas Fire at 25 kW

The propane fire (13 HG) data in Figure 5.7 follows the same trend as the latex fire (04 HG) data. The weighted probability performs better at the beginning (60–95 seconds) with 2.6 degree deviation compared to the temperature heading with a 13.0 degree deviation. In contrast, the temperature heading does better when the robot gets closer to the room with fire (95–128 seconds) with 0.4 degree deviation compared to 10.1 degree deviation from weighted probability. The temperature at the door exceed that of the surroundings and the robot is close enough for the temperature heading to point towards the door. During this period, the weighted probability heading retains a positive heading angle that would lead towards the door on the left. At the very end of the hallway (128–139 seconds) the weighted probability heading performs better (12.9 degree deviation) than the temperature heading (29.4 deviation). The door goes out of the FOV and the robot faces the wall as shown at 206 seconds in Figure 5.7. When faced with the wall, the weighted probability heading points correctly towards the left but the temperature heading points towards the right. The weighted probability heading performs better because it remains mostly positive and it does not have high changes in degrees, as shown in Figure 5.7.

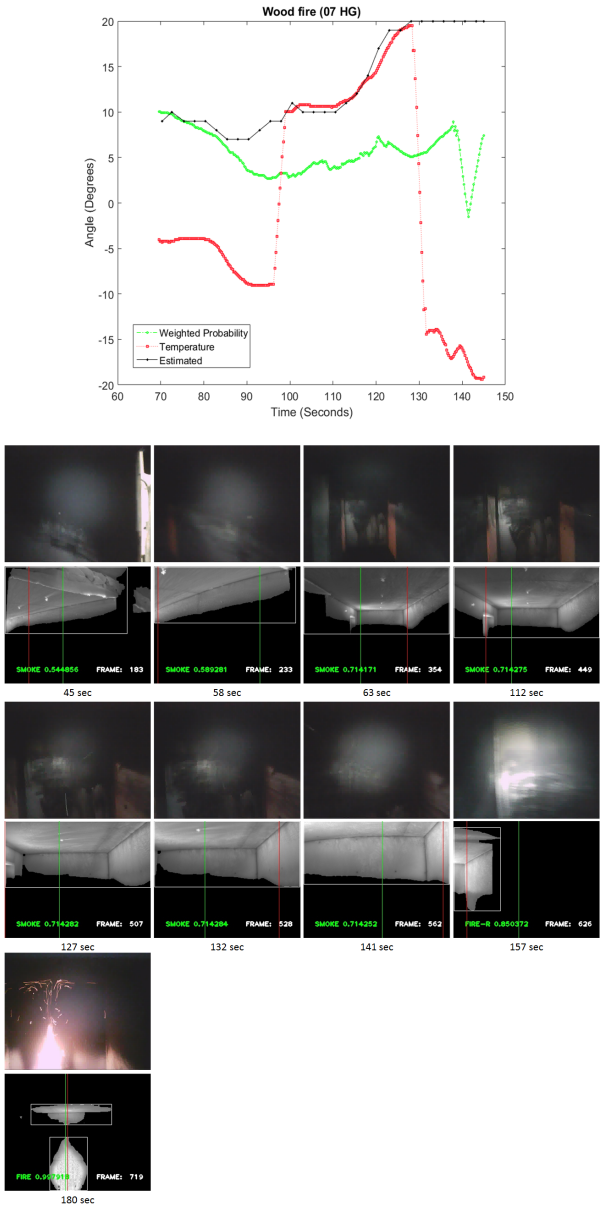

Figure 5.5: Wood crib fire(07 HG) time lapse and headings comparison.

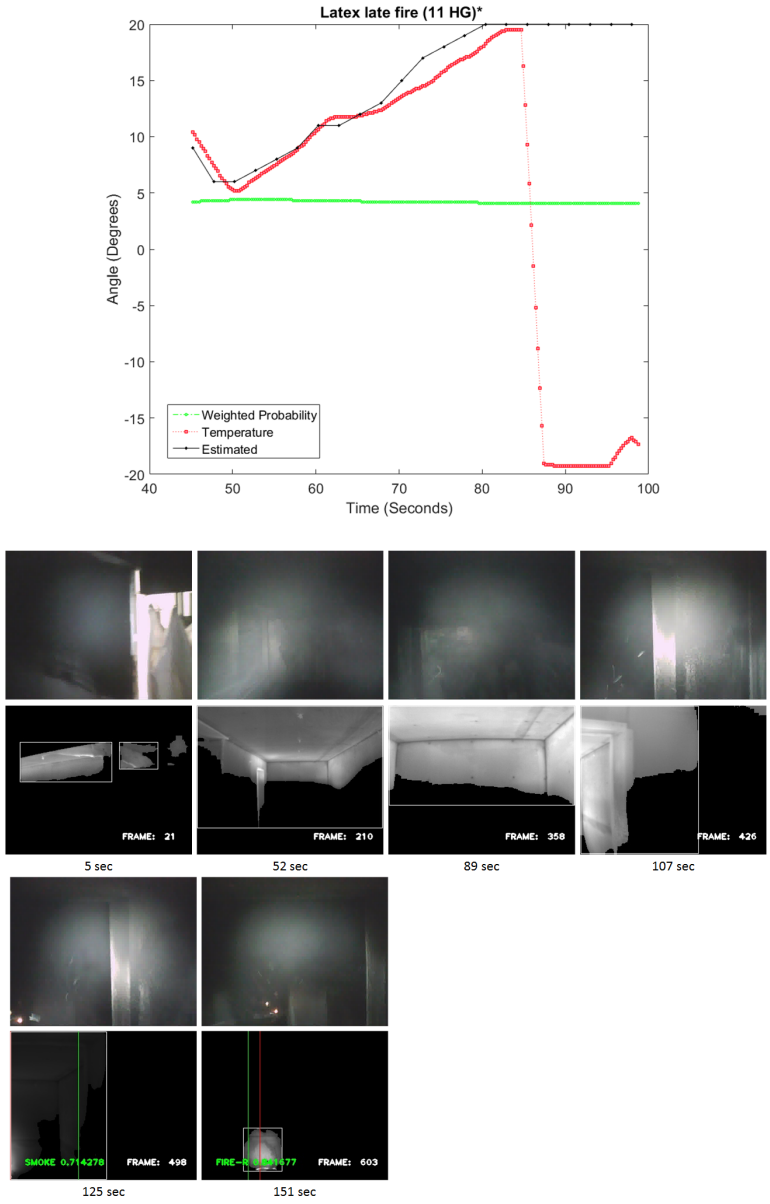

Figure 5.6: Latex late fire (11 HG) time lapse and headings comparison.

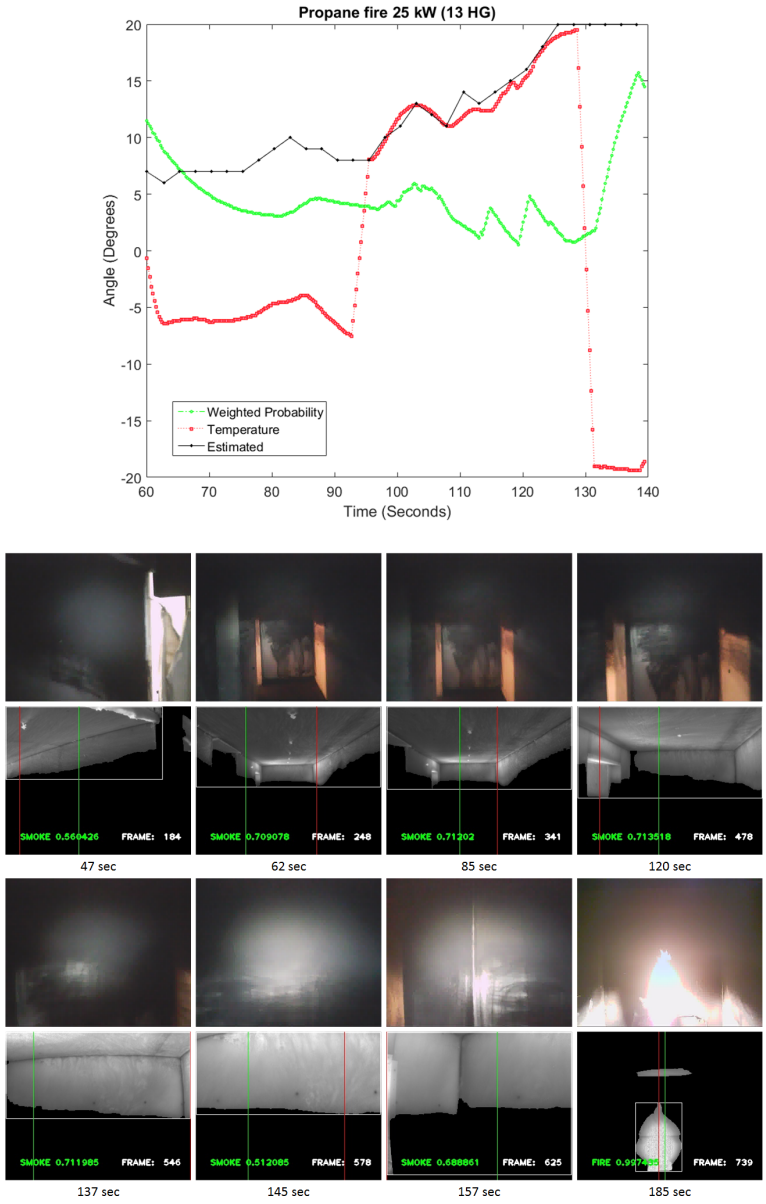

Figure 5.7: Propane fire at 25kW (13 HG) time lapse and headings comparison.

5.2.7 Yaw Heading Angles From (14 HG) Propane Gas Fire at 75 kW

Figure 5.8 contains the propane fire (14 HG) plot and both algorithms do not perform well at the beginning. From 59 to 89 seconds the average deviation for temperature is 15.9 degrees while weighted probability had a 10.8 degree deviation. From 89–107 seconds, the temperature heading consistently points towards the room with fire with a 0.43 average degree deviation and the weighted probability heading focuses on the smoke coming out of the room and has 6.9 degree deviation. Finally, when the robot FOV does not contain the door and faces the wall (107–122) the temperature performs worse (32.4 degree deviation) than the weighted probability (6.4 degree deviation).

At 59 seconds in Figure 5.8, the weighted probability heading begins by pointing towards the room with fire on the left hand side. Then at 77 seconds the weighted probability heading slowly turns towards the accumulation of gases at the other side and does not go beyond. Later, the weighted probability heading increases and follows the correct estimated heading with a 6.9 degree deviation. This behavior shows that the weighted probability can focus on an incorrect section and that it does not make changes quickly.

At 110 seconds, the doorway leading towards the room with fire is not visible and the robot faces the wall. As shown in Figure 5.8 at 122 seconds, the weighted probability algorithm points left but the temperature heading points right. Once again, in an ambiguous situation the weighted probability heading outperforms temperature heading.

5.2.8 Overall Comparison

In order to compare performance, the average degree difference between estimated points and each heading was calculated for the duration of each test. The data in Figure 5.9 indicate that the weighted probability heading performed better in (02 HG), (07 HG), (13 HG), and (14 HG), while the temperature headings performed better in (03 HG), (04 HG), and (11 HG). Therefore, a combination of the two approaches is worth exploring in future work.

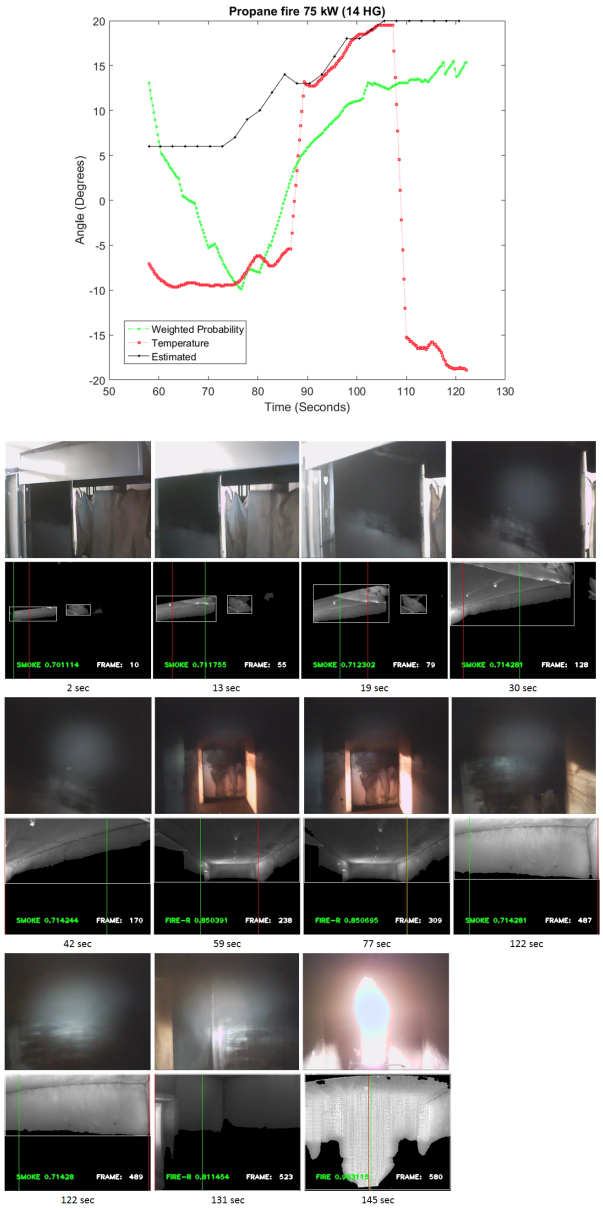

Figure 5.8: Propane fire at 75kW (14 HG) time lapse and headings comparison.

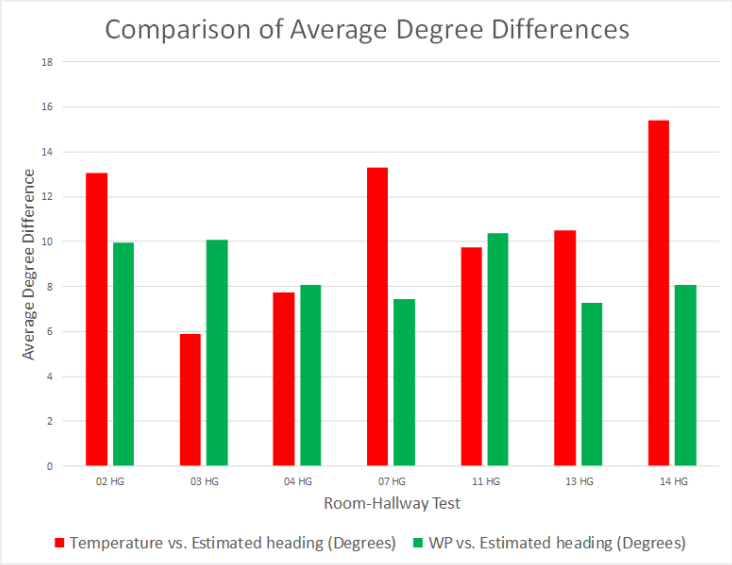

Figure 5.9: Comparison of both methods to the estimated tele-operator degrees.

Chapter 6

Conclusion and Future Work

6.1 Conclusions

Laboratory test results show that the classification method works better in environments where a heated background, such as a wall, can provide thermal information for smoke. In comparison, having no background to detect smoke with IR imaging proves more challenging. The analysis of the classification method highlights the need for more training data to improve classifications accuracy for thermal reflections and possibly fusion of the IR results with RGB.

Simulation tests were used to compare the weighted probability heading to a maximum temperature heading. In addition, comparisons were made to estimated tele-operator heading from training videos. Simulation results show that the maximum temperature algorithm headings led in false directions and were sometimes erratic due to the dynamics of the fire and fire reflections. By weighting the fire environment classes and implementing a Kalman filter, the weighted probability algorithm provided a less erratic heading which would lead the robot toward a fire outside its FOV.

This thesis started with the goal of developing an algorithm that uses classification information to guide a robot to a fire. A robot heading was determined using a weighted probability measurement from fire, smoke, and heat reflection classifications. This heading is determined in each ROI from IR images by finding the highest weighted probability of leading to the fire. ROS (Robot Operating System) enables the algorithm to provide autonomy to more than one type of robotic platform. Both a wheeled platform, named Rene, and a bipedal platform, called ESCHER, are used to test the algorithm in a series of real-life and simulation tests respectively.

When compared to the temperature heading and an estimated robot-operator heading, the weighted probability algorithm does assist the robot with the best direction for it to move

toward the fire. Temperature heading performs better as the robot got closer to the room with fire but it also displayed more erratic behavior. At a distance, the weighted probability heading has a tendency to prioritize smoke and the room with fire may have a lower priority class, such as a hot object. A future fusion of the weighted probability and maximum temperature methods could provide a more robust heading.

6.2 Future Work

Fire causes losses in life, property, and poses an environmental hazard. Robots are explored as a firefighting measure aboard ships in order to reduce reduce these risks. For the application of shipboard firefighting, the limitations imposed by the ship's physical environment and the fire require a real-time system such as the one presented in this thesis. A progression of tests that could follow this research are first, to test in a room-hallway environment using the wheeled platform Rene and second, test the humanoid platform ESCHER at the room-hallway environment using raised sills. Recording the starting point and proprioception data from these new tests would improve the analysis of robot movement and yaw headings. These tests would also help determine the optimal threshold parameter for the weighted probability. Another way to improve analysis is to synchronize the camera images and provide a ground truth to the all classifications. Traditionally IR images are used to see through smoke and with the successful classification of smoke and fire, one can conclude that the classification algorithm works best when a background is present.

Three important modules that must be incorporated alongside the heading algorithm are a higher decision-making system, a collision avoidance system, and a navigation feedback system. The heading provides a way to find a fire but a separate module must veer the robot away from walls and objects it could collide with. The higher decision-making system, such as a state machine, should decide when a heading does not lead towards an open path or what to do when a fire is found. For now, the algorithm simply stops sending movement commands once a fire is found. Finally, a navigation feedback system would help discover how closely the robot followed the path intended and correct for it. Because a single IR camera is necessary for this algorithm, there is also the potential of creating a separate alert system.

In addition, this thesis shows the beginning of a labeled database of IR images which does not currently exist today. Two advantages of having such a database are that it would allow researchers to avoid common machine learning problems, such as over fitting, and it would allow researchers to explore additional machine learning algorithms to detect fire, smoke, and heat reflection. This creates the opportunity to test the algorithm against other types of smoke that have not been tried, for example, hot water vapor. An interesting alternative would be to create artificial samples of fire, smoke, and heat reflections in simulation. The Naive Bayes classifier in this research is a generative classification algorithm that can create new samples; its model is created based on the data it trained on.

For future work, Robot Operating System (ROS) allows for portability the weighted probability algorithm. The DARPA Robotics Challenge held on 2015 proved that robotics can take larger strides by making algorithms easily integrable to robotic platforms. Because the localization algorithm presented in this thesis packaged in a ROS node, it can be open sourced and integrated onto any robotic platform that uses ROS.

Bibliography

[1] Craig H. Shelley. *Marine Firefighting Training and enhancement for Vela Marine International*. 2002. URL: http://apps.usfa.fema.gov/pdf/efop/efo34651.pdf (visited on 10/19/2015).

[2] J. Molis R. Fahy P. LeBlanc. "Firefighter Fatalities in the United States - 2014". In: *National Fire Protection Association* (2015).

[3] Jason Hoevelmann. *Fast and Furious: Primary search tips you should know*. 2011. URL: http://www.firerescue1.com/fire-products/thermal-imaging/articles/1030180-Fast-and-Furious-Primary-search-tips-you-should-know/ (visited on 03/30/2016).

[4] Michael M. Dugan. *How to Properly Search a Fire Building*. 2008. URL: http://www.firefighternation.com/article/firefighting-operations/how-properly-search-fire-building (visited on 03/30/2016).

[5] Alessandro Saffiotti. "Platforms for rescue operations". In: *AASS Mobile Robotics Laboratory. Orebro University, Orebro, Swed* (2004).

[6] LUF GmbH. *LUF Fire-Fighter*. 2015. URL: http://www.luf60.at/ff-luf60-einsatzgebiet/?L=1 (visited on 11/20/2015).

[7] Howe and Howe Technologies. *Thermite*. 2015. URL: http://www.howeandhowe.com/rs1-t2-thermite.html (visited on 11/20/2015).

[8] State Scientific Center Central Research, Design Institute for Robotics, and Technical Cybernetics. *Mobile fire-fighting robot*. 2015. URL: www.neva.ru/CNII-RTC/fireman.html (visited on 11/20/2015).

[9] ARMTEC. *SACI 2.0, System Support Combat Incidents*. 2006. URL: http://v2.armtecbrasil.com/index.php?produto (visited on 11/20/2015).

[10] Pyung-Hun Chang et al. "Control architecture design for a fire searching robot using task oriented design methodology". In: *SICE-ICASE, 2006. International Joint Conference*. IEEE. 2006, pp. 3126–3131.

[11] Teh Nam Khoon, Patrick Sebastian, and Abu Bakar Sayuti Saman. "Autonomous fire fighting mobile platform". In: *Procedia Engineering* 41 (2012), pp. 1145–1153.

[12] Young-Duk Kim et al. "Portable fire evacuation guide robot system". In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE. 2009, pp. 2789–2794.

[13] Dan-Sorin Necsulescu, Jurek Sasiadek, et al. "Fire detection robot navigation using modified voting logic". In: *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*. Vol. 1. IEEE. 2014, pp. 140–146.

[14] MK Rangan et al. "A computer vision based approach for detection of fire and direction control for enhanced operation of fire fighting robot". In: *Control, Automation, Robotics and Embedded Systems (CARE), 2013 International Conference on*. IEEE. 2013, pp. 1–6.

[15] Truc Kim Thi Nguyen and Jong-Myon Kim. "Multistage optical smoke detection approach for smoke alarm systems". In: *Optical Engineering* 52.5 (2013), pp. 057001–057001.

[16] Waqar S Qureshi et al. "QuickBlaze: Early Fire Detection Using a Combined Video Processing Approach". In: *Fire Technology* (2015), pp. 1–25.

[17] J. Kim and B. Lattimer. "Real-time probabilistic classification of fire and smoke using thermal imagery for intelligent firefighting robot". In: *Fire Safety Journal* 72 (2015), pp. 40–49.

[18] Marin Bugarić, Toni Jakovčević, and Darko Stipaničev. "Adaptive estimation of visual smoke detection parameters based on spatial data and fire risk index". In: *Computer vision and image understanding* 118 (2014), pp. 184–196.

[19] Tao Li et al. "An efficient fire detection method based on orientation feature". In: *International Journal of Control, Automation and Systems* 11.5 (2013), pp. 1038–1045.

[20] Amjad D Odetallah and Sos S Agaian. "Human visual system-based smoking event detection". In: *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics. 2012, pp. 840607–840607.

[21] Liqiang Wang et al. "Hybrid fire detection using hidden Markov model and luminance map". In: *Computers & Electrical Engineering* 37.6 (2011), pp. 905–915.

[22] Truong Xuan Tung and Jong-Myon Kim. "An effective four-stage smoke-detection algorithm using video images for early fire-alarm systems". In: *Fire Safety Journal* 46.5 (2011), pp. 276–282.

[23] Magy Kandil, May Salama, and Samia Rashad. "Fire detection using a dynamically developed neural network". In: *ELMAR, 2010 PROCEEDINGS*. IEEE. 2010, pp. 97–100.

[24] Jayavardhana Gubbi, Slaven Marusic, and Marimuthu Palaniswami. "Smoke detection in video using wavelets and support vector machines". In: *Fire Safety Journal* 44.8 (2009), pp. 1110–1115.

[25] Chao-Ching Ho. "Machine vision-based real-time early flame and smoke detection". In: *Measurement Science and Technology* 20.4 (2009), p. 045502.

[26] Byoung Chul Ko, Kwang-Ho Cheong, and Jae-Yeal Nam. "Fire detection based on vision sensor and support vector machines". In: *Fire Safety Journal* 44.3 (2009), pp. 322–329.

[27] Jing Yang, Feng Chen, and Weidong Zhang. "Visual-based smoke detection using support vector machine". In: *Natural Computation, 2008. ICNC'08. Fourth International Conference on*. Vol. 4. IEEE. 2008, pp. 301–305.

[28] Jong-Hwan Kim, Brian Keller, and Brian Y Lattimer. "Sensor fusion based seek-and-find fire algorithm for intelligent firefighting robot". In: *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on*. IEEE. 2013, pp. 1482–1486.

[29] OpenCV. *Image Thresholding*. 2016. URL: `http://docs.opencv.org/trunk/d7/d4d/tutorial_py_thresholding.html#gsc.tab=0` (visited on 03/30/2016).

[30] Liu Jianzhuang, Li Wenqing, and Tian Yupeng. "Automatic thresholding of gray-level pictures using two-dimension Otsu method". In: *Circuits and Systems, 1991. Conference Proceedings, China., 1991 International Conference on*. IEEE. 1991, pp. 325–327.

[31] Luc Vincent. "Morphological area openings and closings for grey-scale images". In: *Shape in Picture*. Springer, 1994, pp. 197–208.

[32] Pedro Felzenszwalb and Daniel Huttenlocher. *Distance transforms of sampled functions*. Tech. rep. Cornell University, 2004.

[33] OpenCV. *Image Segmentation with Distance Transform and Watershed Algorithm*. 2016. URL: `http://docs.opencv.org/3.0-rc1/d2/dbd/tutorial_distance_transform.html` (visited on 03/30/2016).

[34] OpenCV. *Operations on Arrays*. 2016. URL: `http://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html#meanstddev` (visited on 03/30/2016).

[35] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. "Textural features for image classification". In: *Systems, Man and Cybernetics, IEEE Transactions on* 6 (1973), pp. 610–621.

[36] *FLIR A35 f=9 mm*. 2015. URL: `http://80.77.70.144/DsDownload/Assets/73309-0101-en-US.html` (visited on 12/21/2015).