Presence detection and activity recognition using low-resolution passive IR sensors

Troost, M.A.

Award date:
2013

Technische Universiteit
Eindhoven
University of Technology

Presence detection and activity recognition using low-resolution passive IR sensors

by M.A. Troost [0588213]
m.a.troost@student.tue.nl

Date: April 3, 2013
Supervisors: Dr. Oliver Amft, Luis Lopera
Advisor: Frank Clermont

Abstract

Various installations and appliances used by building occupants during the day are manually operated, including office devices and kitchen appliances. Traditionally, activity monitoring systems require multi-modal sensor installations in order to monitor these types of rooms. These large multi-modal installations often require active maintenance to ensure robust operations. Due to large number of different activities performed in such rooms, traditional sensor installations often require a large amount of (different modality) sensors. This greatly increases the complexity of such systems.

The work presented in this thesis introduces a framework which allows for the fine grained detection of objects and user-object interactions from their thermal signatures, using a single low resolution 2D-matrix thermopile. The use of a single sensor greatly simplifies the installation and maintenance of such sensor installations and reduces the interference perceived by the users.

Privacy of the users has been taken into account during the study. As the sensor has a low resolution, typically 40 cm per pixel at 2 m distance, individual users cannot be identified. Also, due to the complexity of the output of the sensor it cannot be interpreted by people who have no knowledge of the context in which the data is obtained. As a result the privacy concerns are similar to those applicable the grid installations of traditional motion sensors.

The presented framework for detecting fine grained user-object interactions consists of three main modules: 1) a sensor layer, 2) an object detection layer and 3) a classification layer. This abstraction provides a high level of flexibility in the framework. In the first module the output of the sensor is conditioned, in the second module objects are detected and tracked based on their thermal signature. In the third module the user-object interactions are classified. Furthermore, this module also classifies the current state of the objects. The state of an object is an indicator for its instantaneous energy consumption.

The framework allows for the classification of 21 activities from a single sensor installment. All 21 activities involve actions commonly performed in a kitchen setting, e.g. activities involving a coffee pot or faucet. The activities are chosen based on their potential for energy savings. To evaluate the performance of the framework two data sets are recorded in the pantry area on floor 3 of the Potentiaal building at the TU/e campus. The first, scripted, data set is used to train and optimize the framework. The validation of the framework is performed on the second data set, which is a real-life scenario, consisting of 5 hours of unscripted activities.

From the evaluation of the framework it is shown that the implemented algorithm is insensitive to small variations in the performed activities. It is also shown that the framework has an excellent performance of 96.4% for activities with a clear thermal signature, e.g. a coffee pot. For appliances for which the activity can only be inferred from circumstantial evidence the performance is relatively low, ranging from 11.4% for detecting opening and

closing of a refrigerator to 45.4% for detecting whether the faucet is off or cold water is being used. The object detection module shows a very good accuracy of 98.2% for detecting people. This indicates that the sensor it very well suited for presence detection and activity recognition for objects which have a clear thermal signature.

Due to the design of the framework, the system is capable of instantaneously, < 1 second delay, providing information about the current activities being performed. Moreover due to its design the framework is capable of processing multi-user scenarios, as the framework performs activity recognition for each user independently.

Acknowledgement

This master thesis project is a cooperation between Eindhoven Technical University and Applied Micro Electronics "AME" BV.

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. Foremost I want to thank Applied Micro Electronics for providing me with the opportunity and means necessary to carry out my master graduation project.

I would like to thank my thesis supervisors Dr. Oliver Amft and Luis Lopera Gonzalez, who have shown a large and consistent interest throughout the project. Our numerous scientific discussions and their many constructive comments have greatly improved this work.

I would like to thank my advisor at AME, Frank Clermont, and all my colleagues at AME for their interest and support in this project.

Finally, I would like to thank Luis for his hard work in reviewing this thesis and turning the results presented into a paper for the SEIT 2013 conference.

Contents

Chapter 1

Introduction

Energy conservation while maintaining occupant comfort is a critical optimization tradeoff in commercial and residential buildings. While modern building energy management systems (BEMS) can control lighting and heating/ventilation systems, various installations and appliances used by occupants during the day are manually operated, including office devices and kitchen appliances. By providing feedback on appliances' usage and energy consumption, occupant awareness on energy needs can be greatly improved. To provide accurate feedback, usage patterns and occupant activities needs to be recognized from ambient sensors.

Ambient sensor modalities that have been successfully used for activity recognition include video cameras and microphones [1, 2]. However, these modalities are often perceived by occupants as privacy intrusive. Moreover, cameras may require regular maintenance to ensure their robust operation. To reduce privacy concerns infra-red motion detectors have been considered for building automation [3, 4].

The goal of this thesis is to develop and implement a framework that is able to detect the presence of people within a designated area. This framework can in turn enable higher level BEMSs to make decisions about energy saving measures. For instance turning off lights and heating when people have left the room or turning off appliances when unused.

First, the report focuses on applying signal analysis in combination with a standard motion detector to create a sensor which is capable of detecting stationary people. Next, a new type of sensor is investigated called a thermopile. Where traditional motion sensors use the pyroelectric effect to detect thermal radiation, thermopiles use the Seeback effect. Because of this effect thermopiles have a linear relation between the absorbed thermal radiation and sensor output. This allows the sensor to continuously detect people within its field of view, even when they have become stationary. Using the thermal signature of stationery objects, it is possible to distinguish objects using a thermopile.

In this thesis a 2-D matrix thermopile is used consisting of 8×8 pixels. At a distance of 2 m the sensor has a resolution of 40 cm. It is shown that with this resolution it is not only possible to detect and identify objects based on their thermal signature, but also to classify the state of the appliance and the user-appliance interaction.

The framework presented in this thesis focuses on 21 activities performed in a pantry setting. The appliances considered during the evaluation of the framework are: a coffee pot / electric kettle, a refrigerator, a faucet and a microwave. Besides these four appliances the framework determines whether a meeting is in progress. The framework provides concurrent responses for all configured and detected objects, thus can inherently handle multi-user

scenarios.

For evaluation of the framework two separate data sets are recorded. One training data set containing scripted activities and one validation data set consisting of a half-day long unscripted recording. The validation data set is used for evaluating the performance of the framework on real-world data. During training optimal feature vectors and parameters are determined for each of the classifiers.

The classification of the activities is performed off-line. An online implementation of the framework is beyond the scope of this project.

The work presented in this thesis has resulted in a paper, which is accepted for publication in the SEIT 2013[1] conference.

1.1 Outline

The remainder of this thesis is as follows. Chapter 2 describes related work in the field of presence detection and activity recognition using passive infrared sensors. Chapter 3 the limitations of the current generation of sensors used in motion detects. Section 3.2 specifically outlines the choices made for selecting a thermopile. Chapter 4 describes the physical prosperities of a matrix thermopile and the resulting limitations for activity recognition. Also an overview of the framework is presented. In chapter 6 and 7 a detailed description is presented of the developed framework. Chapter 8 outlines the performed experiments and evaluation metrics. Chapter 9 outlines the technique used for optimization of the classifiers. The results obtained from running the framework over the validation data set are presented in Chapter 10. Finally, Chapter 11 provides a conclusion and proposes future work on the framework implemented in this thesis.

[1] International Conference on Sustainable Energy Information Technology (SEIT) 2013 - http://cs-conferences.acadiau.ca/SEIT-13/

Chapter 2

Related work

Passive infrared sensor are traditionally separated into two categories. On one hand there are pyroelectric sensors, which due to their physical properties can only detect motion, and on the other hand there are thermopiles, which due to their physical properties can directly measure the temperature of a scene in its field of view.

Traditionally pyroelectric sensors have been used to recognize activities, in [4] they were used to keep track of how many people were in an office room, and in [5] they were used to detect activity as a series of activations of certain areas in the home. Although both presented promising results, the limiting factor is that they both depend on the principle of a gateway and the idea that activities can be modeled solely by entering or exiting an area of coverage. If applied to a constrained location, i.e. a bathroom, a more distributed sensor placement is required; like the one presented in [6]. This approach sometimes requires the placement of sensors in locations where they interfere with the activity being performed. Multiple devices also means that maintenance requires more time, even if the sensors are "tape and forget" [6].

Thermopiles have become, in recent years, considerably more affordable. Furthermore their sensitivity has increased to a level where they have become suitable to detect human presence in typical every day environments. In [7] and [8] it is shown that a thermopile can be used as an effective pedestrian avoidance system. In [9–12] it is shown that using thermal images in tracking and classifying provides superior results compared to CCTV cameras. In [10] it is shown that thermal images provide advantages of images from the visual spectrum for problems like identifying the pose and thus the activity a person might be performing. This is because people are, usually, warmer than the environment. As a result they shine against cooler backgrounds. Thermal sensor are insensitive to luminance variation eliminating detection problem which are caused by changes in sunlight intensity or flickering of fluorescent lighting.

A low resolution line thermopile is used as a gateway sensor in [13] as a replacement for a standard pyroelectric sensor. They show that using a low cost thermopile accuracies can be obtained which are similar to commercial systems which cost 10-25 times more.

As low resolution thermopiles have only recently become affordable enough to be used in large-scale ubiquitous applications, there is no published literature which uses this type of sensor for complex activity recognition on a scale and complexity presented in this thesis.

As thermopiles contain a sensing array, they are closer related to CCTV cameras then to motion detectors. This raises additional privacy concerns compared to motion detectors as it is possible to track objects in the field of view of the sensor. It is however only possible to perform

piecewise tracking in multi-user scenarios as the sensor suffers from tracking ambiguities which are unsolvable for the sensor information alone. For example, if two people occupy the same pixel, or two pixels next to each other, they lose their individual identities. It is possible to classify the detected objects into different categories based on their thermal signatures. However, it is impossible to identify a person. This is because the thermal signature of a person can vary widely between different detection instances. These temperature differences are caused by, for example, the clothes worn by the person or activities performed outside the field of view of the sensor. As a single person occupies typically only one or two pixels, the resolution of the sensor is too low to perform robust identification like face or posture detection. Unlike data obtained from CCTV camera, data obtained from a grid thermopile cannot be interpreted by people who have no knowledge of the context in which the data is obtained. This reduces possible privacy risks if the system is compromised.

As the thermopile is similar in appearance to that of the traditional motion sensors, which are already ubiquitously install in, amongst other places, office areas, the same level of intrusion is perceived by its installment by the end users.

Chapter 3

Initial work

Traditional passive infra-red sensors are extensively used in office buildings to reduce the energy consumption for example by switching of lights in empty conference rooms. Due to their extensive use these types of sensors can be easily sourced for a low price (typically less than one euro). However due the physical property of the pyroelectric effect these sensors have one big limitation, they are unable to detect stationary objects. This is a direct result of the pyroelectric effect, which is only sensitive to changes in the amount of absorbed thermal radiation and not the amount of radiation itself.

A large amount of research is performed to improved presence detection using PIR sensors [14–17]. These papers however assume a purely abstract binary sensor model. As a result of this abstraction many inherent sensing modality limitations are ignored. For example the bursty nature of positive detections and the inability to detect stationary object resulting in numerous false negatives [18]. An explanation for this behaviour is given later.

If this type of PIR sensors are used in a network it becomes possible to extract higher level information from the scene, such as counting the number of people present and localizing their position. The accuracy of this greatly depends on the resolution of the grid and the number of sensors used [19]. Independent of the configuration of the grid, PIR sensors are at most only capable of piecewise tracking. This is because they suffer from ambiguities which cannot be solved from the binary state of a single sensor alone. For example, when two people cross paths their individual identities are lost. A standard method to bypass this limitation is to make, often unrealistic, smoothness assumption about paths taken [18].

Most standard PIR sensor based methods use geometric formulations to determine the path of a person from the intersection of the different sensing areas inside a grid [19]. In [19] a sensor node is constructed with multiple PIR sensors arranged in a circle and pointed radially outwards from the nodes center. They show that by using this type of sensor it is possible to detect the location of a person. In case multiple of these sensors are used [20] shows that it is possible to track the path of a moving person. However, stationery objects where ignored in this study.

The main disadvantages of pyroelectric sensors as traditionally used in motion sensors are [18]:

1. They cannot detect objects who have become stationary after initial motion, resulting in numerous false negatives.

2. Their output is highly bursty, resulting in many separate detection for a single person.

These disadvantages are largely ignored by the vast majority of PIR-based research by limiting their system to single-person scenarios and/or assuming people are always moving. A standard technique used to compensate for the bursty output of the sensor is to use a *refractory period*, which is a kind/type of dead-time/blind-time. All detections which fall within the *refactory period* of an earlier event are ignored [18], as they are assumed to be triggered by the same object. The standard technique to compensate for the inability to detect stationary objects is to require a minimal amount of motion within a time frame. This technique however is unable to differentiate between persons who are stationary and those who have actually left the field of view of the sensor [18]. Also, the system is remains unable to detect people who remain stationary for a time-period longer than the used time frame.

Several experiments are performed to determine the suitability of using a standard PIR sensor for not only motion detection, but true presence detection. These results of these experiments are discussed in detail in Chapter A. A short summary of the obtained results is presented in Section 3.1/ The goal of the performed experiments is to determine if it is possible to detect, using a single sensor, the presence of a single person which remains stationary for extended periods of time within the view of the sensor.

3.1 Experiments

Traditionally a large number of high-pass, low-pass filters and amplifiers are used to interface the sensing elements of the sensor with the other hardware for further processing [21, 22]. This has a clear effect on the output of the sensor. The typical output obtained in such a configuration is shown in Figure 3.1a. Here a person moves with constant speed through the field of view of the sensor. For this experiment a common type of PIR sensor is used, which contains two active areas. These two active areas result in two peaks which are clearly visible in the output signal. One peak represents when the person is within the field of view of the first active area and the other represent when the person is in the field of view of the second active area. Due to the internal wiring of the active areas the two peaks have opposite polarity.

To increase the sensitivity of the sensor a Fresnel lens is commonly used [22]. A Fresnel lens divides the field of view of the sensor into distinct zones, each containing a projection of the active areas of the sensor. As a result, if a person moves through the field of view of the sensor the waveform shown in Figure 3.1a is repeated multiple times, once for each of the projections from the lens. This is also the reason for the bursty output of the sensor. The output of the sensor in combination with a Fresnel lens is shown in Figure 3.1b.

In Figure 3.1b the output of the sensor is also shown in case the object becomes stationary half way through the field of view. It is shown that within 1.5 seconds after the object becomes stationary, it becomes indistinguishable from no object present at all. The exact time it takes before the object becomes indistinguishable depends on the corner frequencies of the filters used. The results shown in Figure 3.1 matches those presented in [20] and [18]. This means that any information concerning presence, which might be present in the signal, is removed by the filters. To further investigate the suitability of the sensor these filters are removed.

The step response of the sensor, with all filters removed, is shown in Figure 3.2. The results obtained matches the theoretical model of a PIR sensor. Using the following equation

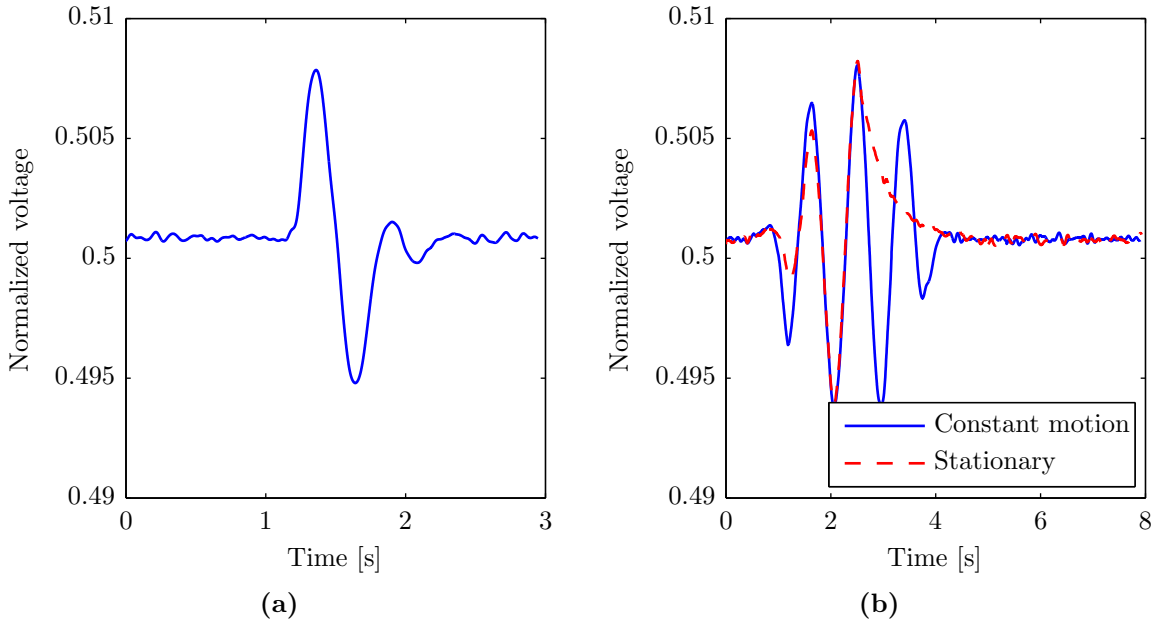


Figure 3.1: Normalized PIR sensor output for when (a) no lens is used and (b) a Fresnel lens is used. Shown in red is the output for an object which becomes stationary after initial motion.

the model is specified [20, 22]:

$$S = S_0[1 - e^{-\frac{t}{\tau_e}}][(1 - C)e^{-\frac{t}{\tau_t}} + C] + b \tag{3.1}$$

here τ_e and τ_t are the electrical and thermal time constants respectively, C is the ratio between the peak out voltage and the steady-state offset, b is bias voltage and S and S_0 are the sensor output and the stead-state response respectively. The steady-state offset C is caused by imperfections in the sensing elements themselves, e.g. electrical and thermal leakage. In Figure 3.2 it is shown that $C \approx 0$. As a result the difference between the offset of the output voltage for no object and a stationary object becomes indistinguishably small. This makes it impossible to use standard, cheaply available, components to detect presence using this steady-state offset.

The exponential decay caused by τ_t could be used to detect object which have become stationary after initial motion. From Figure 3.2 it is estimated that this technique would allow for at most 1.5 minutes of presence detection. After this period the exponential decay will become indistinguishable from the bias drift of the sensor, which is $\leq 25\,{}^{\mu V}/_{s}$. This short time period provides no advantages over the method of requiring a minimal amount of motion within a time frame.

3.2 Discussion

The standard PIR sensors are, due to their physical properties, unable to directly detect presence. The results obtained in Section 3.1 show that using additional signal analysis it is

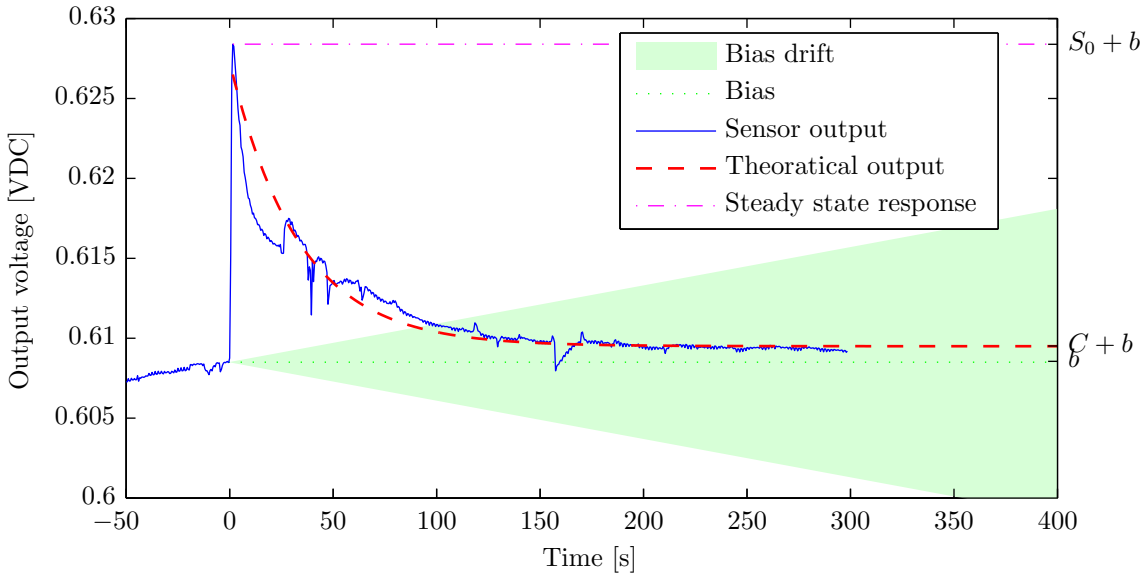


Figure 3.2: Step response for a PIR sensor.

possible to perform short term presence detection for at most 1.5 minutes. This is however only valid in single-person scenarios. If multi-person scenarios are used the output of the sensor becomes increasingly more complex, making it impossible to detect the exponential decay.

As an alternative to the traditional PIR sensors a new type of passive infra-red sensor has recently emerged in the field of energy reduction, which is called a *thermopile*. These sensors rely on a different physical effect as the traditional PIR sensors. Thermopiles have a linear relation between the detected temperature and the output of the sensor [22]:

$$S \propto T \tag{3.2}$$

here S is the sensor output and T is the average temperature of the scene. As a result these type of sensors are suitable to detect stationary objects. Another major advantage of these sensors is, that they are available in integrated grid configurations. This eliminates the need to build custom grids if object tracking is desired.

As a result of this build-in grid and ability to detect stationary object a great amount of information can be obtained from the sensor. Compared to the traditional sensors not only true presence can be detected, but also the number of people present, their current location and their trajectory. Using this information of location and trajectory additional information can then be inferred from the scene like, for example, the activities currently taking place.

This increase in available information results in that a thermopile is $20 - 30\times$ as expensive as a single traditional PIR sensor. For an 8×8 grid version however, it offers an enormous amount of information more than the traditional sensor. A custom grid with the similar functionality containing traditional PIR sensors, only lacking the ability to detect static objects, are $3 - 4\times$ as expensive as a thermopile grid sensor. In applications where this additional information is useful the increase in obtained information greatly outweighs the possible price difference.

For the remained of this thesis an 8×8 thermopile is used. The output of this sensor is used to perform activity recognition, aimed at energy consumption reduction in a multi-user setup.

Chapter 4

Theory

This chapter first discusses the physical properties of a thermopile compared to PIR sensors which are extensively used in motion detectors. This section concludes with the resulting limitations of a thermopile for activities recognition. Finally an overview of the developed framework is presented. The framework is presented in detail in chapter 6 and 7.

4.1 Physical properties' thermopile

Thermopiles are sensors capable of measuring the thermal radiation absorbed on their active area. They belong to the category of thermal detectors which generate a small thermoelectric voltage proportional to the detected radiation. The main difference between a thermopile and a pyroelectric sensor, as used in traditional motion detectors, is the physical phenomena which is used to detect the amount of incoming thermal radiation.

The operation principle of a pyroelectric sensor is based on the polarization of a crystal as response to a thermal flow. However, under steady-state conditions, free-charge carriers neutralize the polarized charge [22]. As a result the output of a pyroelectric sensor is linear with the derivative of the detected thermal radiation [22]. This type of sensors can therefore not be used to detect scenes which are in steady state, as their output contains no information for these scenes [22]. Steady-state scenes are not necessarily static scenes, they can be dynamic scenes as long as the total amount of thermal radiation absorbed by the sensor remains the constant.

Thermopiles use the Seebeck effect as their operation principle. The Seebeck effect describes the electric current in a closed circuit composed of two dissimilar materials when their junctions are maintained at different temperatures. The voltage developed can be approximated using the following formula [22, 23]:

$$V = (S_B - S_A)(T_h - T_c) \tag{4.1}$$

where S_A and S_B are the Seebeck coefficients of the two materials and T_h and T_c are the temperatures of the hot and cold junctions respectively. From Equation 4.1 it can be derived that the developed voltage only depends on the instantaneous temperatures. The output of a thermopile is thus independent of the state path of the system and only depends on the current state. This type of sensors are therefore well suited to detect the state of a system even in steady-state.

In order to increase the sensitivity of thermopiles, multiple thermocouples are place in a grid and connecting electrically in series [22, 24]. By varying the number of thermocouples used in a thermopile the desired sensitivity level is obtained [22]. The output voltage of a thermopile is given by the sum of the voltages developed over the individual thermocouples:

$$V = (S_B - S_A)(T_{1,h} - T_{1,c}) + \ldots + (S_B - S_A)(T_{N,h} - T_{N,c}) \tag{4.2}$$

$$= (S_B - S_A) \sum_{i=1}^{N} [T_h(i) - T_c(i)] \tag{4.3}$$

where N is the number of junctions connected in series. If it is assumed that all cold junctions are maintained at same (ambient) temperature, equation 4.3 can be simplified to:

$$V = \Delta S \left(\sum_{i=1}^{N} T_h(i) - N T_{amb} \right) \tag{4.4}$$

where ΔS is the difference in Seebeck coefficients of the two materials, $T_h(n)$ is the temperature of the hot junction as a function of its index and T_{amb} is the ambient temperature. If $N \to \infty$, the sum over N will become an integral over the domain D, where D is the area of the sensing element.

$$V = \Delta S \iint_D (T(x, y) - T_{amb}) \, \mathrm{d}x \, \mathrm{d}y \tag{4.5}$$

where $T(x, y)$ is the temperature at location (x, y) on the hot junction. If the sensor uses a mapping $m : V \mapsto T_s$, where V is the voltage over all thermocouples and T_s is the absolute average temperature of the scene, the output of the sensor can be calculated using the following equation:

$$T_s = \frac{\iint_D T(x, y) \, \mathrm{d}x \, \mathrm{d}y}{\iint_D 1 \, \mathrm{d}x \, \mathrm{d}y} \tag{4.6}$$

The output of the sensor, T_s, is thus the average temperature perceived by this active area.

All objects in the field of view (FOV) of the sensor, which are not hidden by other objects, contribute to the output of the sensor [22,23]. If it is assumed that all object have homogenous constant temperature, a transmittance of $\tau = 0$ and an emissivity of $\epsilon = 1$, meaning the objects behave like a perfect black-body and are completely opaque, Equation 4.6 can be simplified to:

$$T_p(u, v) = \frac{\sum_{i=1}^{N} T_{obj}(i) \left([A_{obj}(i) \cap A_{pixel}(u, v)] \setminus \vartheta(i) \right)}{A_{pixel}} \tag{4.7}$$

where N is the number of objects that are in the FOV of the sensor, $A_{obj}(i)$ is the polygon of the projection of the i^{th} object onto the sensor, A_{pixel} is the polygon describing the pixel area and $\vartheta(i)$ is the polygon which indicates the part of $A_{obj}(i)$ that is occluded by other objects. If the objects are sorted in increasing z-index, so that $A_{obj}(1)$ and $A_{obj}(N)$ is closed and furthest to the sensor respectively, the occlusion polygon can be calculated using the following equation:

$$\vartheta(i) = \bigcup_{j=1}^{i-1} A_{obj}(j) \tag{4.8}$$

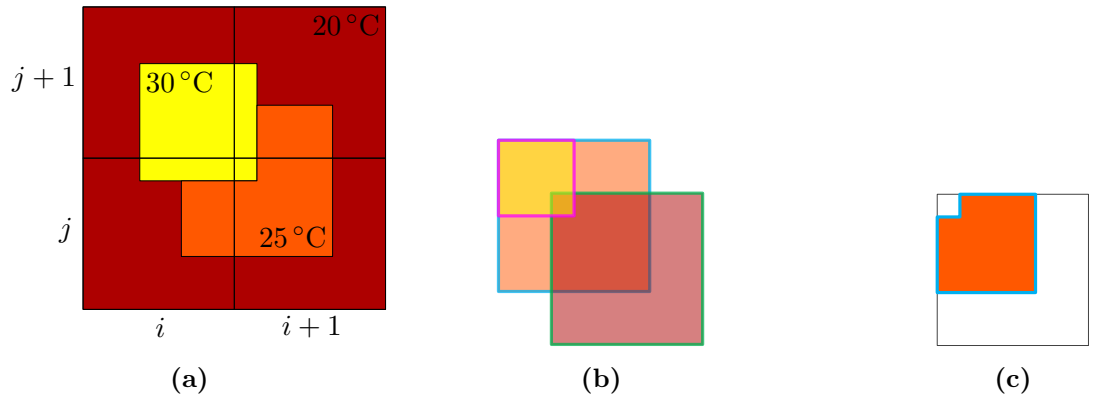


Figure 4.1: Projection of object 2, 25 °C, from scene onto pixel $(i + 1, j)$; (a) original scene, (b) $A_{obj}(2)$ shown in blue; $A_{pixel}(i+1, j)$ shown in green; and $\vartheta(2)$ shown in magenta, (c) projection onto pixel.

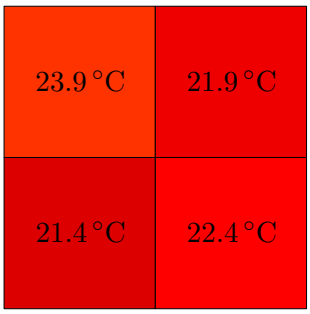


Figure 4.2: Theoretical thermopile output for Figure 4.1a.

An example of Equation 4.7 using three objects, $T_{obj}(1) = 30\,°\text{C}$; $T_{obj}(2) = 25\,°\text{C}$ and $T_{obj}(3) = 20\,°\text{C}$, is shown in Figure 4.1. The sensor output for the scene shown in Figure 4.1a is shown in Figure 4.2.

4.1.1 Limitations

The behaviour described in section 4.1 results in the following limitations when the sensor is used for activity recognition.

- Depending on the mounting position of the sensor, a person may positions himself in such a way in the FOV of the sensor, that the activity is not perceived. An example would be a person who's back is facing the sensor will obscure all activities performed in front of him.
- Two objects of the same temperature placed next to each other are perceived by the sensor as a single object with an area equal to the union of the two objects. An example would be furniture. As the temperature of the furniture is equal to the ambient temperature it is invisible to the sensor.

- As mentioned in [25], for objects with a low emissivity ($\epsilon \ll 0.9$) reflections become a dominant part of the temperature perceived by the sensor. This creates the illusions of objects which aren't actually present. An example is a coffeepot on a stainless steel counter top. Stainless steel has an emissivity of $\epsilon = 0.075$. As a result the perceived temperature of the counter top will consist for 92.5% of reflections. Due to this low ϵ the coffeepot will make the counter top appear much warmer than it is.

Chapter 5

Framework

The developed framework will use a matrix thermopile consisting of 8×8 pixels to classify a set of different activities. The activities which are classified are outlined in Section 5.1. An overview of the top-level architecture of the algorithm in given in Section 5.2.

5.1 Activities

To detect and classify the different activities, their distinct thermal images are used. By detecting regions in the frame which have different temperature than the background, objects are spotted. Each detected object gets a state associated with it. The state of an object is defined as its status in the current frame. Each detected object has also an interaction associated with it to each of the other detected objects. An interaction is defined as the level of influence two objects have to change each others state. The combination of a state and interaction forms an activity.

The activities investigated are based on the state of common household kitchen appliances and the user interactions with them. An overview of the appliances investigated and their associated states and interactions is given in Table 5.1. Apart from the activities performed involving one of the four appliances, also a meeting activity performed between two people is included. Two people are considered to be performing a meeting if they result in a single hot spot (blob) in the thermal image. This does not necessarily imply that the two objects are inseparable from each other in the thermal image. It only implies the distance between them is smaller than the resolution of the sensor. A watershed [26] type of algorithm could be used to separate the two individual persons from the single resulting blob. The precise definitions used for the classification of the different states are:

- On/Off; a binary state indicating if the appliance is consuming energy.
- Off/Cold/Hot; a ternary state applicable only to the faucet. The state indicates not only if water is being consumed, but also the temperature of the water.
- Open/Closed; a binary state applicable only to the refrigerator. The state indicates the status of the refrigerator door. The appliance itself is always consuming energy, however from the state of the door the amount can be determined.
- Yes/No; a binary state indicating if a meeting is in progress between two or more persons.

Table 5.1: Set of activities to be classified.

	Coffee pot	Faucet	Microwave	Refrigerator	Meeting
State	Off On	Off Cold Hot	Off On	Closed Open	—
Interaction	Away Present Serving	Away Present	Away Present	Away Present Interacting	No Yes

The precise definition used for the different interactions are:

- Away/Present; a binary interaction indicating the possibility that a person is using the appliance. A person is "Present" if he is close enough to interact with the appliance. From the possible state change of the appliance it can then be determined if an actual interaction is taking place.
- Serving; a refinement of the "present" interaction indicating a person is moving the carafe of the coffee pot and is most likely serving a cup of coffee. For the coffee pot the "present" interaction indicates a person standing next to the coffee pot but not moving the carafe.
- Interacting; a refinement of the "present" interaction indicating a person is possibly changing the state of the refrigerator door. For the refrigerator the "present" interaction indicates a person is close enough to the refrigerator to take objects out but not close enough to open the door.

5.2 Architecture

By making the framework homogenous for all activities a high level of flexibility if obtained. This is desirable as it allows for easy adding or removing of activities. To obtain this homogenous layout the framework is divided into two main modules, namely:

1. Object detection; this module is responsible for conditioning and analyzing the raw output of the sensor. The output of this module are all the objects present in the current frame, including their complete history up until the current frame. To help with this analysis, a map containing the location of the appliances is used.
2. Interaction classification; this module classifies the state and interaction between all objects detected using the object detection module.

The two modules are executed in sequential order for each frame.

Section 5.2.1 and 5.2.2 give an overview on the architecture of the two different modules. Chapter 6 and 7 give an in-dept analysis of all processing steps performed in the object detection and interaction classification module respectively.

5.2.1 Object detection

An overview of the architecture of the object detection module is shown in Figure 5.1. The four steps involved in the detection process are:

1. Lens correction. For accurate classification it is important for an activity which is performed in one part of the frame to be identical in appearance to the same activity performed in another part of the frame. Due to barrel distortion and the mounting angle of the sensor this is not the case in the raw sensor output. See Section 6.1 for the error induced by these distortions. To compensate for these distortions the sensor output is multiplied with the inverse of the distortion to create a rectangular homogenous grid, where feature appearance is independent of the location inside the frame.

2. Occupancy map. Using the corrected sensor output an occupancy grid is created. In the occupancy grid the most probable state, occupied or free, for each pixel is determined. A pixel is considered occupied when the temperature difference between the pixel and the ambient temperature is above a threshold.

3. Object detection. From the output of the occupancy map objects are created using 8-connectivity connected-component labeling. Separate objects are created from pixels identified as warmer and colder than ambient temperature respectively. As a result an object can not be partly warmer than and partly colder than the ambient temperature. Using the map, which contains prior knowledge of the location of the appliances, all appliances which are not detected in the current frame are added. The failure to detect an appliance is a result of this state. For example when the faucet is turned "off" it has a similar thermal signature as the background.

4. Object tracking. In order to determine the history of an object it is tracked across multiple frames. The current state and the history of the object will be used in the classification module.

An example output of the first step of the object detection module is shown in Figure 5.2. Here a scene is analyzed with two dynamic objects present. One near a hot static object in the lower right corner, shown in red and aqua respectively, and one in the center left part of the image, shown in green. Figure 5.2a shows the sensor output after lens correction is applied. In Figure 5.2b the pixels which are classified as occupied are outlined in green. In this frame one pixel, at $(1, 1)$, is incorrectly classified as occupied. At this location a table is located which has, as it is an inert object, the same temperature as its surrounding. However due to the difference in emissivity between the table top and the floor, the sensor perseveres the table form having a different temperature.

Figure 5.2c shows the output of the object detection module. The bounding boxes of all objects detected using the occupancy map are shown in green, whereas all objects which are added using a priori map information are shown in blue. Figure 5.2d shows all objects labeled with a unique color.

5.2.2 Interaction classification

An overview of the architecture of the interaction classification module is shown in Figure 5.3. The classification module uses the result of the object detection module to determine the most

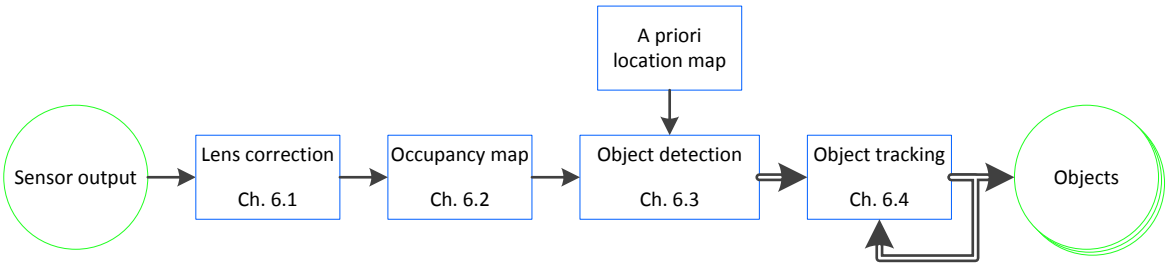


Figure 5.1: Architecture object detection module.

Table 5.2: Object classification types.

Class	Type
Static	Coffee machine Faucet Microwave Refrigerator
Dynamic	Single person Multiple persons

probable states of and interactions between the detected objects. The first step of the module is to classify each object as either a static or a dynamic object. Static objects are all objects of which their location is known a priori. All remaining objects are dynamic objects. All objects classified as static are then labeled as the corresponding appliances. All objects classified as dynamic are then labeled as either being a single or multiple persons. At the output of this step all objects are classified as one of the six possible types specified in Table 5.2.

The next step of the module is to classify the states and interactions of all objects. Each object is classified independently to determine its state. Next all dynamic objects are cross classified against all static objects to determine their interaction. As a result of this each static object has n interactions associated with it, one for each dynamic object. A separate state-machine is used per object to filter and limited the allowed state and interaction transitions. For example, a cup of coffee cannot be served if nobody was present at the coffee pot in the previous frame. In order to reduce the output noise, objects which have a lifetime shorter than the filter window size used in the state-machine are not considered for the output of the current frame. After processing all interacting with the state-machines, each static object has k interactions associated with it, where $1 \leq k \leq n$. It is assumed that all static objects only support single-user interactions. An Activity of Interest Selector (AIS) is used to create a mapping from the set of k interactions to a single interaction.

An example of the output of the classification module, continuing on Figure 5.2, is shown in Table 5.3. Each of the 7 detected objects is first classified as one of the object types, shown in Table 5.3a. The colors used to identify the objects are the same as in Figure 5.2d. Table 5.3b shows the state of each object after applying the state-machine. As shown, the blue noise pixel is not considered in the output as its lifetime is too short. Table 5.3c shows

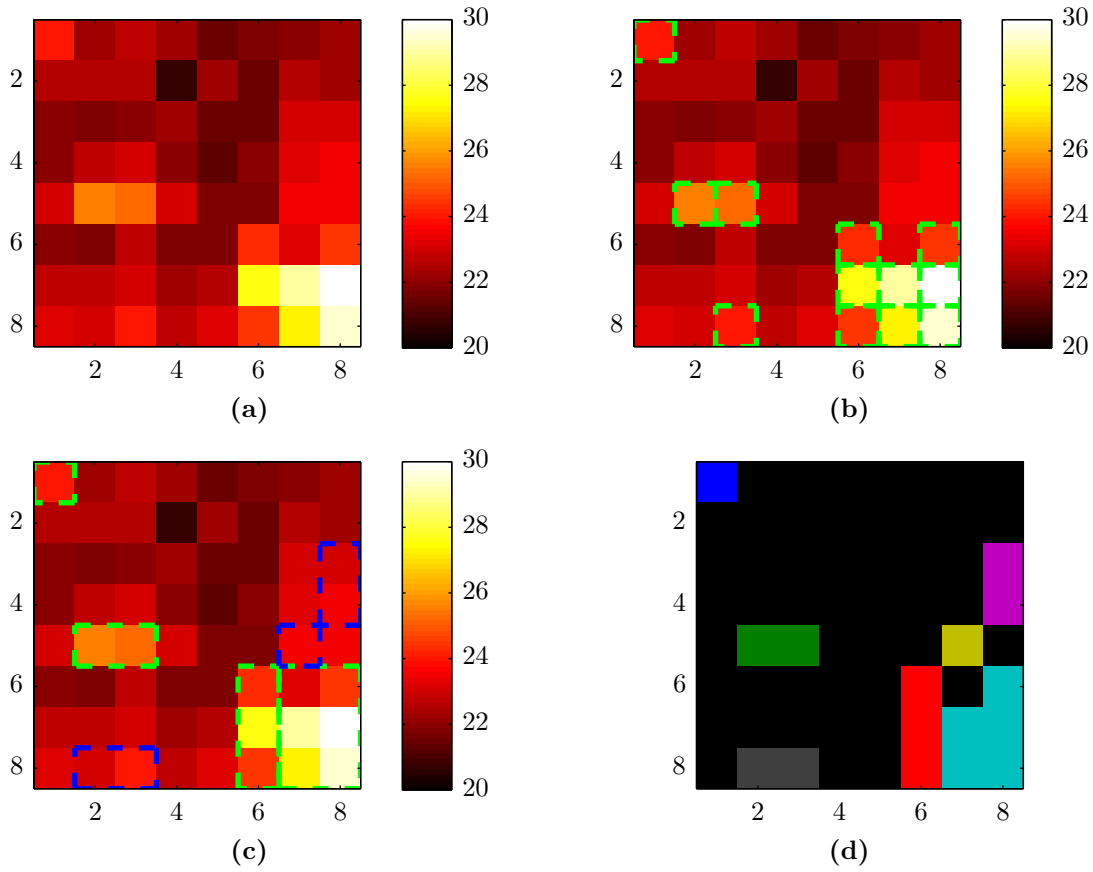


Figure 5.2: Output object detection module for a scene with two dynamic and four static objects; (a) lens corrected sensor output, (b) resulting occupancy map with occupied pixels outlined in green, (c) detected objects (green) and added objects (blue), (d) resulting labeled objects - red, green and blue are dynamic object and magenta, yellow, aqua and gray are static objects.

the interaction classification between each static and dynamic object. Underlined are the resulting output interactions selected by the AIS. For the coffee pot the resulting interaction is "serving" as it is the activity of interest in this case.

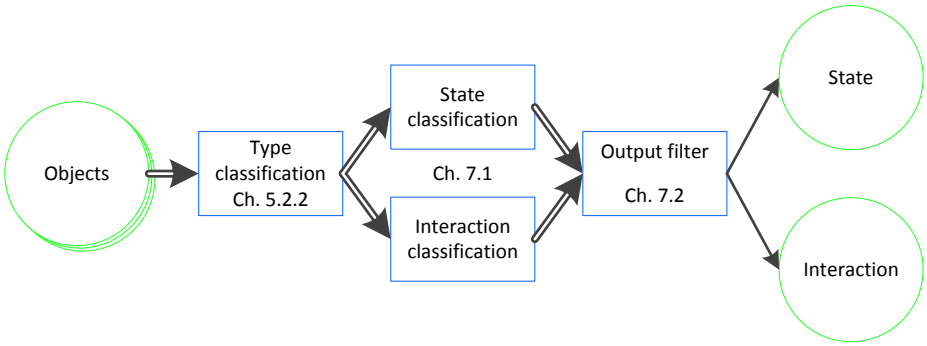


Figure 5.3: Architecture interaction classification module.

Table 5.3: Output of the interaction classification module based on Figure 5.2; (a) shows the object type, (b) shows the object state and (c) show the object interaction with the resulting output interactions underlined.

(a)

	Type
■	Single person
■	Single person
■	Coffee pot
■	Faucet
■	Microwave
■	Refrigerator
■	Single person

(b)

	State
Person 1	No
Person 2	No
Coffee pot	On
Faucet	Off
Microwave	Off
Refrigerator	Closed

(c)

	Person 1	Person 2
Coffee pot	Away	Serving
Faucet	Away	Away
Microwave	Away	Away
Refrigerator	Away	Away

Chapter 6

Object detection module

This chapter provides a detail outline and design choices for the algorithms implemented in the object detection module.

6.1 Lens correction

Due to the low cost lens used in the sensor it suffers from poor optical performance. Most importantly the image suffers from a high degree of barrel distortion. Barrel distortion is caused by the decrease of image magnification with the distance from the optical axis. Figure 6.1a shows the projection of the pixel array at 2.2 m. Indicated with circles is the central angle of each pixel, the outline of the pixels is drawn at their half angles. Note that the figure shows the pixel projections, as a result the distortion in the image is the inverse of that shown in Figure 6.1a. So, as the pixel projection shows pincushion distortion and as result the image will show barrel distortion. The barrel distortion of the image is shown in blue in Figure 6.1b.

The radial distortion is corrected using Brown's distortion model. Brown's model is given by the following formula [27]:

$$r_c = r_u + K_1 r_u^3 + K_2 r_u^5 \tag{6.1}$$

where r_c and r_u are the corrected and uncorrected distance of the pixel with respect to the optical axis respectively. K_n are the radial distortion coefficients, here $K_1 = 7.4 \cdot 10^{-3}$ and $K_2 = 0.17 \cdot 10^{-3}$ are used. These values of K_n are obtained by optimizing the mean square error of the real pixels center positions and the desired rectangular grid such that Equation 6.1 is minimized. The pixel center positions (central angles) are specified in [24] (Chapter 9, Optical properties). These are the same angles as shown in Figure 6.1a.

Figure 6.1 shows the fitting of the uncorrected pixel centers to a rectangular grid. Shown in blue is the uncorrected image, here the barrel distortion can be clearly visible. The corrected pixel centers are shown in red. It is shown that after correction the pixel occupy a homogenous grid. The black arrows show the vectors of the correction function for each pixel center.

Equation 6.1 is used to warp the raw sensor output into an undistorted image. Figure 6.1 shows the result of the warping. Shown in red is the distortion correction function and shown in green is the resulting warped output. It is shown that the output after warping is a homogenous grid.

Due to the lens distortion and mounting of the sensor the projected area is not constant for all pixels. This has as result that the perceived temperature of an object differs with its

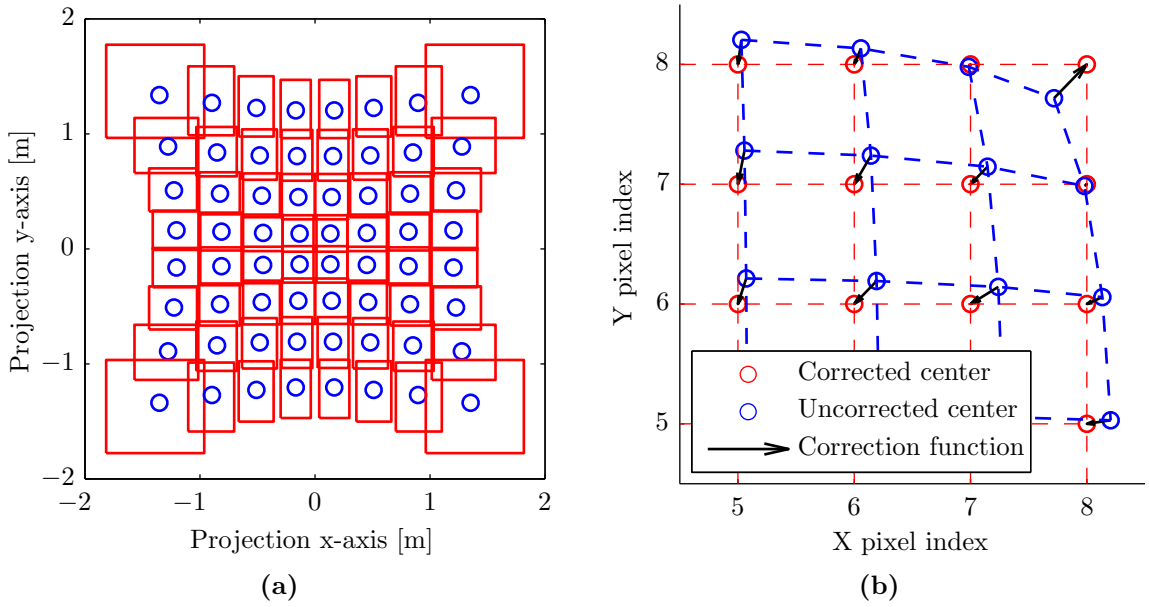


Figure 6.1: Projection of pixel array at $d = 2.2\,\text{m}$; (a) uncorrected pixel projection showing pincushion distortion, (b) distortion frame shown in blue and the resulting corrected frame shown in red. Correction function is shown as black arrows.

location inside the field of view. This is because the projection ratio between the background and the object is different for each pixel. This property can also be derived from Figure 6.1a. An object which would fill an entire pixel in the center of the field of view would only fill $\approx 15\%$ of a pixel in one of the corners. As a result of Equation 4.7, this means that the temperature elevation perceived by the sensor for one of the corner pixels is only $\approx 15\%$ of the elevation the sensor would perceive for the same object in the center of the FOV.

This problem of difference in projected area between pixels is not solved with barrel correction, as this only warps the pixels and does not actually take into account the area of a pixel. This is because barrel correction is originally intended for images taken in the visible spectrum, where the pixels contain different information then in the infra-red.

To solve the problem of different sized projection between pixels, the projected area of each pixels is normalized. Normalization of the area is performed by dividing the uncorrected project area of a pixel by the theoretical projected area of pixel if a perfect lens would be used. The temperature of a pixel after normalization of the area is given by the following equation:

$$T_c = CA_u\Delta T + T_{amb} \tag{6.2}$$

$$C = \frac{1}{4d^2 \tan\Theta^2} \tag{6.3}$$

where T_c is the corrected temperature of a pixel, ΔT is the temperature increase or decrease of a pixel, T_{amb} is the ambient temperature, A_u is the uncorrected projected area of a pixel [24] and C is the area normalization constant. Here the normalization constant, C, equals the area of the theoretical pixel projection at distance d with viewing angle Θ. The viewing

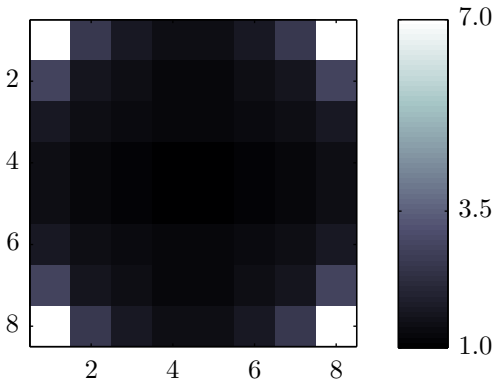


Figure 6.2: Area correction, CA_u, due to lens distortion; with $\Theta = 8\,^\circ$, $d = 2.2\,\text{m}$ and A_u from Figure 6.1a.

angle, Θ, for the used sensor is $8\,^\circ$ [24]. The result of CA_u is shown in Figure 6.2. It is shown that the correction factor for the four corner pixels is very large ($7\times$). This makes these pixels unusable for detecting small temperature differences as their variance is increased $49\times$ compared to the pixels in the center of the image, due to this area normalization. For the remaining 56 pixels the variance is increased, on average, $1.04\times$. This has no noticeable negative effect on the detection capabilities of these pixels.

6.2 Occupancy grid mapping

6.2.1 Interpretation

From the sensor a measurement matrix $\mathbf{z}$ containing the temperature of the individual pixels is obtained each cycle.

$$\mathbf{z} = \begin{pmatrix} T_{1,1} & \cdots & T_{1,N} \\ \vdots & \ddots & \vdots \\ T_{N,1} & \cdots & T_{N,N} \end{pmatrix} \tag{6.4}$$

It is assumed that all measurements in $\mathbf{z}$ are i.i.d., meaning the probability that a pixel contains motion is independent of the remainder of the scene.

$$\forall t \in \mathbf{z}\,[P(t|\mathbf{z} \setminus t) = P(t)] \tag{6.5}$$

A mapping is made from the measurement space $V = \{x : 4x \in \mathbb{Z}\}^{N\times N}$ to the decision space $D = \mathbb{R}^{N\times N}$ by means of a sensor model which is generated a priori. Here $N \times N$ is the total number of pixels of the sensor. As all measurements are i.d.d., the sensor model satisfies the following mapping function [7].

$$f : V(u, v) \mapsto D(u, v) \tag{6.6}$$

The decision space D is represented through a grid m where each cell $m_{u,v}$ contains a probabilistic value. This grid has the same shape as the measurement grid, namely $N \times N$. In

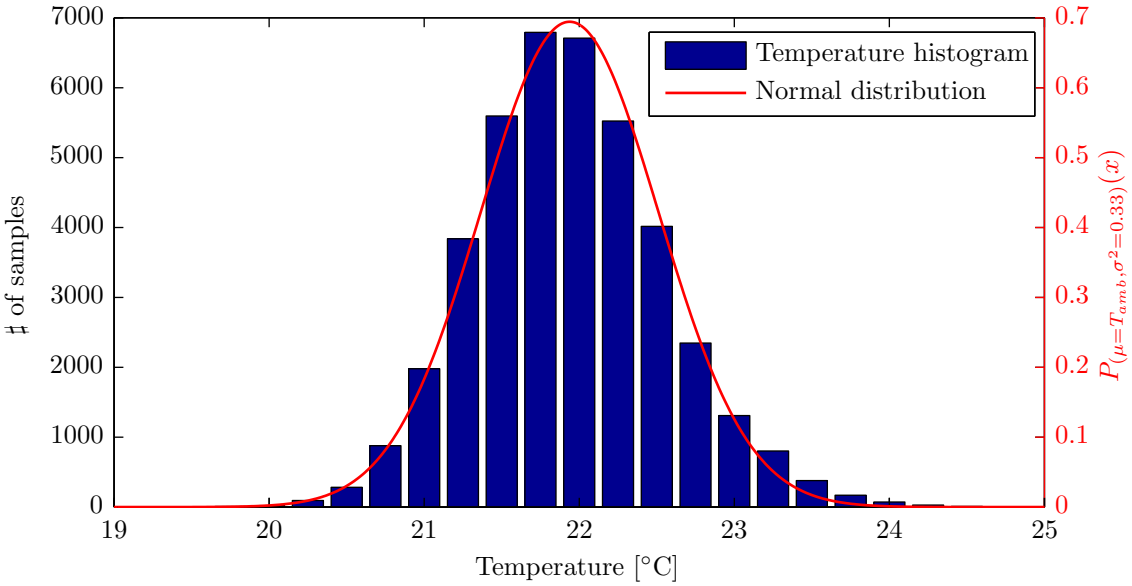


Figure 6.3: Histogram of 1 minute ($n = 40860$) of empty scene (blue) and normal distribution with $\mu = T_{amb}$ and $\sigma^2 = 0.33$ in (red).

probabilistic interpretation methods, a sensor model contains a representation for the sensor's interaction with the scene, and outputs a probability distribution $P(o|t)$ of the physical property [7]. Here o represents the event of an object present and t represent the event temperature T is measured. Using Bayes theorem the equation can be written as:

$$P(o|t) = \frac{P(t|o)P(o)}{P(t)} \tag{6.7}$$

Figure 6.3 shows a histogram created from 40860 samples (1 minute of movie) with no objects present in the scene (blue) and a normal distribution fitted to the histogram (red). 1 minute of movie is used to reduce the error induced by changes in ambient temperature. From the experiments the performed in Chapter 8 probability $P(T|\overline{o})$ is determined to have normal distribution with mean $\mu = T_{amb}$ and variance $\sigma^2 = 0.33$. The PDF for $P(T|\overline{o})$ is then:

$$P(t|\overline{o}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-T_{amb})^2}{2\sigma^2}} \tag{6.8}$$

As the used measurement space V is discrete the continuous PDF from Equation 6.8 is transformed into a discrete PDF using the CDF. For this the property is used that the sensor rounds all measurements to the nearest 0.25 K and thus, the probability of a measurement equals the integration of $P(t|\overline{o})$ over all temperatures rounded to that particular measurement value. The resulting discrete PDF $P'(t|\overline{o})$ is given by the following equation:

$$P'(t|\overline{o}) = F(t + 0.125|O = \overline{o}) - F(t - 0.125|O = \overline{o}) \tag{6.9}$$

$$F(t|O = \overline{o}) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{t - T_{amb}}{\sqrt{2\sigma^2}}\right)\right] \tag{6.10}$$

where $F(t|O = \overline{o})$ is the conditional CDF of $P(t|\overline{o})$. Figure 6.4 shows $P'(t|\overline{o})$ in relation to the histogram of Figure 6.3.

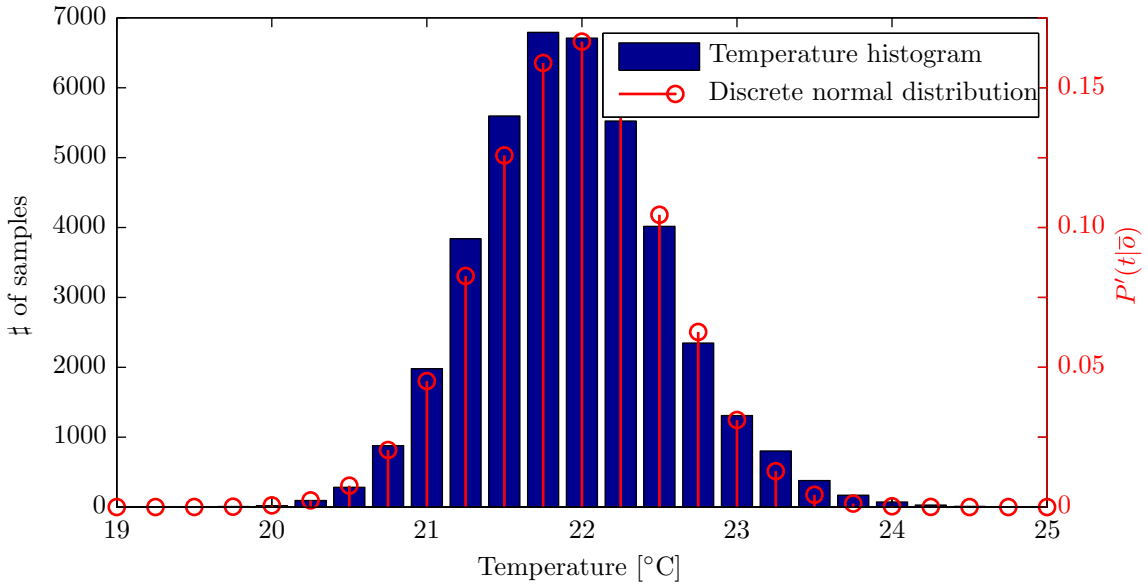


Figure 6.4: Histogram of 1 minute ($n = 40860$) of empty scene (blue) and discrete probability distribution of $P(t|\overline{(o)})$ (red).

The probability that an object is present given a certain temperature is calculated in an analog fashion. As the temperature of the object is unknown, we are interest in objects of all possible temperatures. A uniform PDF is used to calculate $P(t|o)$, as it is assumed that all temperatures are equiprobable to occur for an object. The discrete PDF $P'(t|o)$ is given by Equation 6.12.

$$F(t|O=o) = \begin{cases} 0 & \text{for } t < T_{min} \\ \frac{t-T_{min}}{T_{max}-T_{min}} & \text{for } t \in [T_{min}, T_{max}] \\ 1 & \text{for } t \geq T_{max} \end{cases} \tag{6.11}$$

$$P'(t|o) = F(t+0.125|O=o) - F(t-0.125|O=o) \tag{6.12}$$

where $[T_{min}, T_{max}]$ is the temperature domain of all possible objects. Combining equations 6.7, 6.9 and 6.12 the probability for no object present given a measured temperature becomes:

$$P(\overline{o}|t) = \frac{P'(t|\overline{o})P(\overline{o})}{P(t|\overline{o})P(\overline{o}) + P(t|o)(1-P(\overline{o}))} \tag{6.13}$$

The probability for a cell occupied then becomes:

$$P(o|t) = 1 - P(\overline{o}|t) \tag{6.14}$$

The mapping function f from Equation 6.6 can be described by the following relation [7, 28]:

$$D(u,v) = P(o|V(u,v)) \tag{6.15}$$

Figure 6.5 shows the influence of $P(\overline{o})$ on $P(\overline{o}|T)$. It is shown that for large $P(\overline{o})$ the probability of an object with a temperature of T_{amb} becomes smaller. Figure 6.6 shows the mapping function f for a scene with one object.

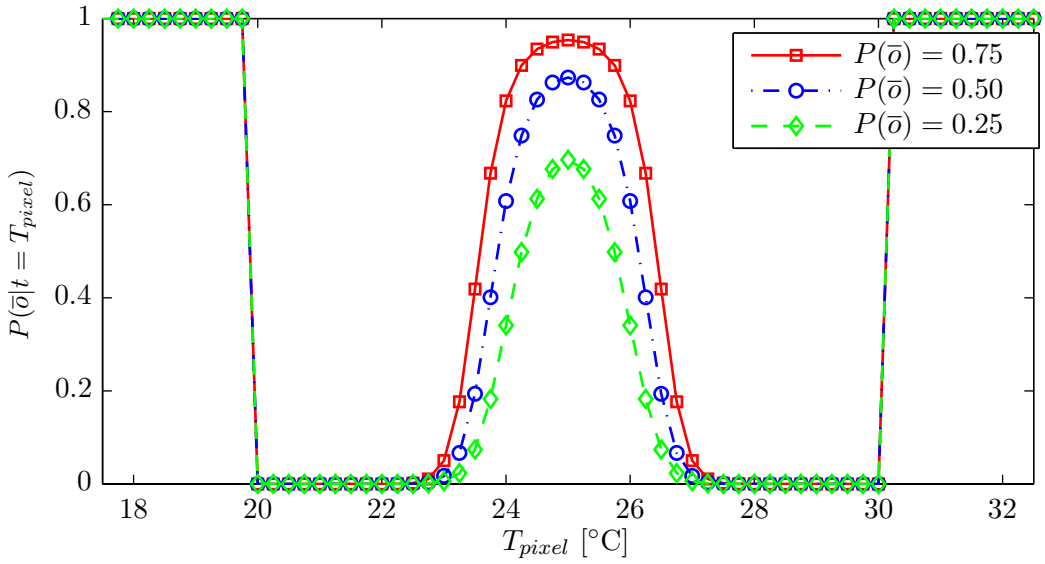


Figure 6.5: Probability of no object present given a measured temperature T_{pixel} and $P(\overline{o})$. $T_{amb} = 25\,°\text{C}$ and $20\,°\text{C} \leq T_{obj} \leq 30\,°\text{C}$. Red is $P(\overline{o}) = 0.75$, blue is $P(\overline{o}) = 0.5$ and green is $P(\overline{o}) = 0.25$.

The probability $P(\overline{o})$ is determined through experiments in Chapter 8 to be 0.95. For the experiment the complete training set, containing 30 minutes of movie, is analyzed and the number of pixels occupied by objects divided by the total number of pixels.

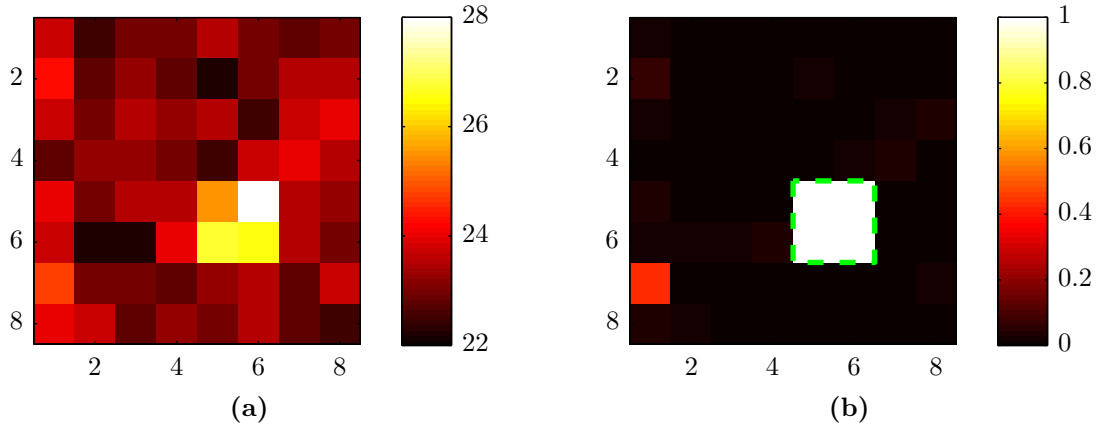


Figure 6.6: Thermal image and occupancy probability map of a scene with one object; (a) scene with $T_{obj} = 27\,°\text{C}$ and $T_{amb} = 23\,°\text{C}$, (b) resulting occupancy probability map for $P(o) = 0.95$ and $20\,°\text{C} \leq T_{obj} \leq 30\,°\text{C}$, object outlined in green.

6.2.2 Integration

Using the probabilities from the decision space D possible location of objects are determined. As the occupancy map is generated from a dynamic scene traditional mapping algorithms will not work. For example [7, 28, 29] all use a priori knowledge about a cell being occupied. An example is the log odds' representation used in [7] where I_t is the log odds of a cell being occupied.

$$I_t = \log\left(\frac{P(o|t_t)}{1-P(o|t_t)}\right) + \log\left(\frac{1-P(o)}{P(o)}\right) + I_{t-1} \tag{6.16}$$

All these equations require, depending on the history, many iterations before a cells state is changes state. However in a dynamic scene this is undesirable as cells can change state every iteration. [8] outlines a method where prior map knowledge and motion estimation is used to overcome these limitations. The motion estimation implies that instead of using the cells own history, the history of the cell which is predicted to move to the cell during the current measurement cycle is used. The downside of this technique is that the motion vectors need to be accurately estimated. To overcome this problem [8] assumes the concept of occupancy preservation. In the concept of occupancy preservation the number of cells occupied at t must be equal to the number of cells occupied at $t-1$. If more cells are occupied, the least probable ones will be discarded. This is undesirable, as in our scene objects can appear and disappear between scenes.

Instead it is assumed that all frames are independent, meaning $P(o_t|o_{t-1}) = P(o_t)$. This is an oversimplification of the reality, however due to the low resolution of the sensor it is arguably correct. Due to the resolution accurate motion estimation is impossible. Figure 6.7 shows the gradient of the motion vector for 200 (dynamic) objects extracted from the training data set. The average motion vector length of an object is 0.25 pixels with a maximum of 1 pixel. The probability a pixel becomes occupied due to the motion of an object in the previous frames is given by the following formula:

$$P(o_t(u,v)|n_{t-1}(u,v) = true) = n\sum_{i=1}^{9}\frac{i}{9}B(i, 0.05) = 0.135 \tag{6.17}$$

The binomial distribution calculates the probability a pixel has i occupied neighbouring pixels in the previous frame. This is multiplied with the probability an occupied neighbouring pixels moves onto the current pixel, which is $\frac{1}{9}$. The notion of at least one neighbour is defined by Equation 6.18.

$$n_t(u,v) = \exists(p,q) \in \mathbb{N}\left(|p-u| \leq 1 \wedge |q-v| \leq 1 \wedge o_t(p,q) = true\right) \tag{6.18}$$

where (u,v) is the pixel in the current frame and (p,q) is the pixel in the previous frame. The difference between $P(o_t)$ and $P(o_t|n_{t-1} = true)$ is very small (0.05 vs. 0.135), thus justifying the independence between frames assumption.

Using this independence assumption the mapping from decision space D to occupied pixel is made using a binary maximum a posteriori probability (MAP) detector. A MAP detector chooses the most likely hypotheses amongst all possible options. As a result it minimizes the expected number of errors [30]. In this case there are two hypotheses:

$$H1 : O = false \tag{6.19}$$

$$H2 : O = true \tag{6.20}$$

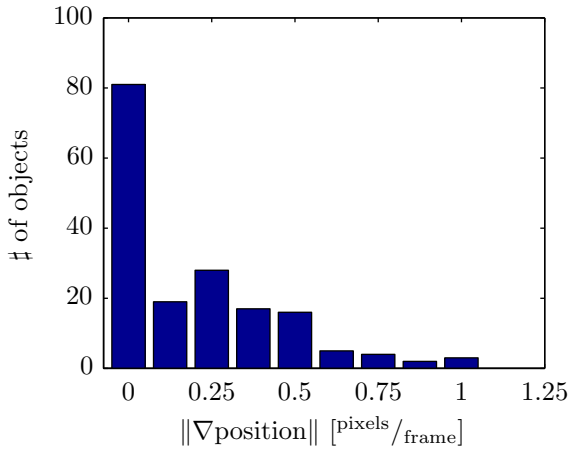


Figure 6.7: Histogram of the magnitude of the motion vector for 200 objects extracted from the training data set.

Table 6.1: MAP detector threshold.

$P(o)$	Threshold [K]
0.05	1.84
0.135	1.63

For the MAP detector equation 6.13 and 6.14 are used. The threshold for the detector is given by [30]:

$$O = \begin{cases} true, & \text{if } L(o) \geq \tau_{MAP} \\ false, & \text{otherwise} \end{cases} \tag{6.21}$$

where τ_{MAP} and $L(o)$ are defined as:

$$\tau_{MAP} = \frac{P(\overline{o})}{P(o)} \tag{6.22}$$

$$L(o) = \frac{P(t|o)}{P(t|\overline{o})} = \frac{1 - P(t|\overline{o})}{P(\overline{o})} \tag{6.23}$$

Table 6.1 shows the threshold for choosing $H2$ over $H1$. The threshold here is defined as the absolute difference between the temperature of the pixel and T_{amb}. The table also shows that the threshold changes 0.21 K if motion estimation would be used. As the resolution of the sensor is only 0.25 K using motion estimation would effectively changes nothing to the output of the MAP detector and thus the ability to detect objects.

6.3 Object detection

From the output of the occupancy map individual objects are formed. Objects are formed by applying connected-component labeling on the output of the MAP detector. As connectivity,

8-connected neighbourhood is used. 8-connected neighbourhood is chosen over 4-connected neighbourhood as small objects moving diagonally across the scene will result in two diagonally connected pixels, which would result in two separate objects if a connectivity of 4 is used. A downside of this high degree of connectivity is that it allows for objects with very low solidity, meaning it has a tendency to form very large objects consisting of only a few occupied pixels. Another downside, which is both applicable to 4-connected and 8-connected neighbourhood connectivity, is its ability to form object outlined by complex polygons. A complex polygon is defined as being neither convex nor concave, meaning it either insects itself or it has a boundary consisting of discrete circuits (e.g. a hole inside the polygon). This is undesirable as, due to the low resolution of the sensor, all objects of the size we are interested in can be outlined by simple polygons. Both disadvantages are a result of the algorithms' preference to create as large as possible objects.

To reduce the complexity of the classification it is desirable to create objects which closely match the basic components of the scene. A scene consisting of 3 objects should result in 3 detected objects, not in 1 detect object consisting of all 3 objects combined. This is directly opposite to the preference of the algorithm to create as large as possible objects. Two changes are made to the standard labeling algorithm to overcome its disadvantages.

First, the occupancy map is segmented into two separate maps, one containing all pixels warmer than ambient and one containing all pixels colder than ambient temperature. The goal of this segmentation is to prevent the formation of objects which are partly warmer and partly colder than ambient. This property of objects being either homogeneously warmer of homogeneously colder than ambient holds for all objects of interest to the framework. The two objects which are capable of changing their temperature between warmer and colder than ambient are the faucet and the refrigerator. However as these two object both only occupy a single pixel each, they are also perceived by the sensor has having a homogenous temperature.

This following reasoning shows that using the ambient temperature as segmentation boundary minimizes the number of errors made in the formation of objects. Given two objects, with a temperature of $T_1 = T_{amb} + T$ and $T_2 = T_{amb} - T$ respectively. The probability distribution for the temperature measured by the sensor is, based on Equation 6.8, expressed by the following formulas:

$$P(t|O = T_1) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{[t-(T_{amb}+T)]^2}{2\sigma^2}} \tag{6.24}$$

$$P(t|O = T_2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{[t-(T_{amb}-T)]^2}{2\sigma^2}} \tag{6.25}$$

A pixel would belong to object T_1 if $P(t|O = T_1)P(O = T_1) \geq P(t|O = T_2)P(O = T_2)$. As states in Section 6.2, the probability of an object, $P(O)$, is equal for all temperatures, so $P(O = T_1) = P(O = T_2)$. The threshold for choosing the warm object over the cold object is given by the following equations:

$$P(t|O = T_{amb} + T) \geq P(t|O = T_{amb} - T) \tag{6.26}$$

$$\frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{[t-(T_{amb}+T)]^2}{2\sigma^2}} \geq \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{[t-(T_{amb}-T)]^2}{2\sigma^2}} \tag{6.27}$$

$$\tag{6.28}$$

This simplifies to:

$$e^{-\frac{[t-(T_{amb}+T)]^2}{2\sigma^2}} \geq e^{-\frac{[t-(T_{amb}-T)]^2}{2\sigma^2}} \tag{6.29}$$

$$-\frac{[t-(T_{amb}+T)]^2}{2\sigma^2} \geq -\frac{[t-(T_{amb}-T)]^2}{2\sigma^2} \tag{6.30}$$

$$[t-(T_{amb}+T)]^2 \leq [t-(T_{amb}-T)]^2 \tag{6.31}$$

$$-2tT + 2T_{amb}T \leq 2tT - 2T_{amb}T \tag{6.32}$$

$$t \geq T_{amb} \tag{6.33}$$

where T_{amb} is the ambient temperature, T is the temperature of an arbitrary object and t is the temperature of the pixel under investigation. This means that all pixels measured as warmer than ambient also belong most likely to an object warmer than ambient. This proves that segmenting the occupancy map using the ambient temperature as a threshold minimises the expected number of errors made in the formation of objects.

The second change made to the algorithm is to split the hot and the cold occupancy map once more using a priori known object locations. This prevents the creating of objects of which it is known that they should consist of, at least, two objects. All static objects are contained in map containing a priori knowledge about their location. This has as advantage that for the interaction classification the detected objects can be split into two categories, namely static objects and dynamic objects. The category of static object will consist of all objects which are created using the a priori map knowledge and the category of dynamic objects will consist of all other objects.

The connected-component labeling algorithm [31] used consists of following four steps:

1. Initialize all pixels in the unlabeled state.
2. Search for the next unlabeled and occupied pixel.
3. Use flood-fill to label all connected components.
4. Repeat steps 2 and 3 until all occupied pixels are labelled.

The flood-fill algorithm used in step 3 uses a receive implementation using the following five steps:

1. If pixel is not occupied return.
2. If $\text{sign}(T_{cp} - T_{amb}) \neq \text{sign}(T_{pp} - T_{amb})$ return.
3. Label pixel using object index.
4. Perform flood-fill on each of the eight connected components.
5. Return.

Here T_{cp} and T_{pp} are the temperatures of the pixel considered to be labeled and the pixels already labeled respectively.

6.3.1 Invisible objects

In some states, some objects have a thermal signature which is similar to the background or ambient temperature. For example the coffee pot or faucet in the off state. In this state these objects will be invisible to sensor and thus they are not detected during the object detection phase. To overcome this limitation a a priori map of known objects is used to supplement the output of the object detection module with all missing known objects.

6.3.2 Ambient temperature estimation

The ability of the occupancy map algorithm to distinguish objects from background heavily depends on the accuracy of the ambient temperature estimation. The output of the sensor contains besides the temperatures measured by the pixel array also the temperature of a thermistor connected to the sensor housing. However, due to self-heating the temperature of the sensor housing is higher than the actual ambient temperature. An advantage of the thermistor is its high precision, compared to the measurement taken from the pixel array. The variances are $\sigma_t = 7.08 \cdot 10^{-4}$ for the thermistor reading and $\sigma_p = 0.01$ for the pixel readings which are averaged per frame.

To get an accurate estimation of the ambient temperature a filter must be used. The general properties of a suitable filter would be its ability to remove high frequency noise present on the pixel readings, while still accurately tracking ambient temperature variations. It is also preferred that the filter only exhibits short startup transients and that it is insensitive for large errors in its initial value estimations.

For the ambient temperature estimation a Kalman filter is selected, which combines both the pixel array and the thermistor reading. A Kalman filter estimates the offset of the high accurate thermistor reading using the noisy pixel temperature readings. This is an ability which cannot be obtained if a simple IIR filter over the noisy pixel data would be used. A disadvantage of the Kalman filter is that it assumes that there is a relation between the temperature variations measured by the thermistor and the ambient temperature. The prediction and correction equations for the Kalman filter are [32]:

$$\hat{x}_{t|t-1} = A\hat{x}_{t-1|t-1} + Bu_t \tag{6.34}$$

$$P_{t|t-1} = AP_{t-1|t-1}A^T + Q \tag{6.35}$$

$$K_t = \frac{P_{t|t-1}H^T}{HP_{t|t-1}H^T + R} \tag{6.36}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(T_t - H\hat{x}_{t|t-1}) \tag{6.37}$$

$$P_{t|t} = (I - K_tH)P_{t|t-1} \tag{6.38}$$

where A is the state transition model, B is the control input model, H is the observation mode and Q and R are the process and measurement noise covariance respectively. A simple model is used where it is assumed that the ambient temperature remains constant during the entire measurement period, $A = 1$. As the thermistor and pixel array measure the same unit of measurement, $B = 1$. As the output of the sensor contains no scaling, $H = 1$.

To reduce the influence of the presence of objects on the ambient temperature estimation only pixels which are indicated as not occupied in the MAP detector output are used. If all pixels are indicated to be occupied the previous prediction is reused and the a posteriori error

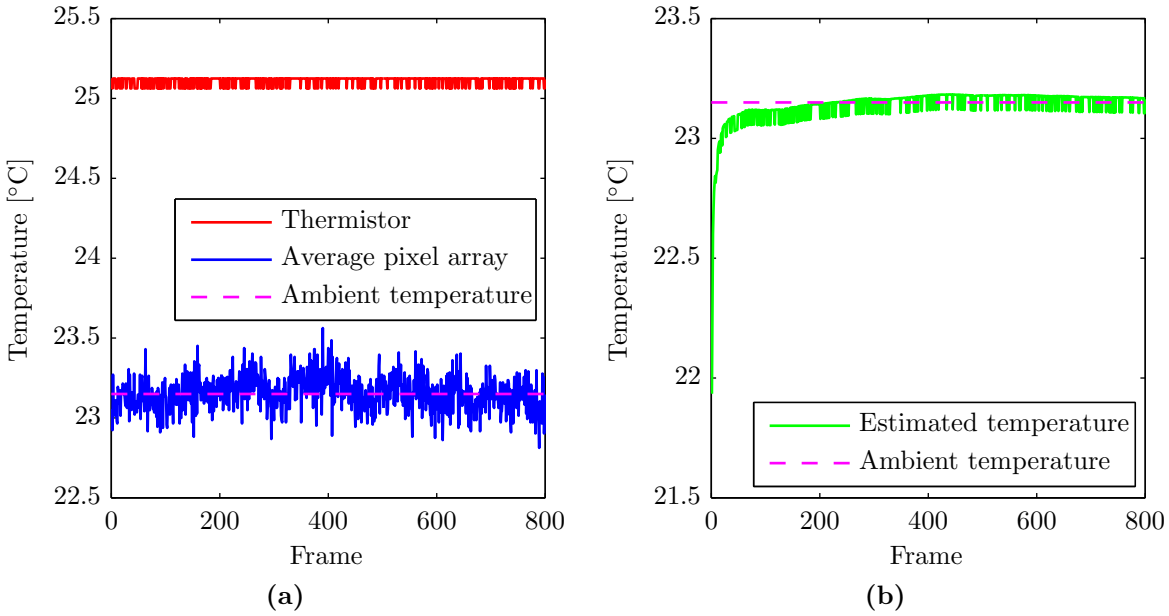


Figure 6.8: Ambient temperature estimation for $T_{amb} = 23.1\,°\text{C}$; (a) thermistor output (red) and average temperature per frame (blue), (b) estimated temperature using Kalman filter. Initialized at $T_0 = 22\,°\text{C}$ and $P_0 = \sigma^2$

. Actual ambient temperature is shown in magenta.

covariance matrix $P_{t|t}$ is not corrected using Equation 6.38. The measurement used in the filter, T_t is calculated using the following equation:

$$T_t = \begin{cases} H\hat{x}_{t|t-1}, & \text{if } \mathbf{z}_t(1 - \mathbf{map}_t^T) = 0 \\ \frac{\mathbf{z}_t(1-\mathbf{map}_t^T)}{\|\mathbf{z}_t\|}, & \text{otherwise} \end{cases} \tag{6.39}$$

where $\mathbf{z}_t$ is the vector containing all pixel temperatures and $\mathbf{map}_t$ is the vector containing a 1 or a 0 for all occupied and free pixels respectively. As the minimum number of pixels used in the filter is 1 the measurement noise covariance R is chosen to be equal to the variance of a single pixel, $\sigma_M^2 = \sigma^2 = 0.33$. From the training set the process noise covariance Q is estimated, using minimum mean square error (MMSE), to be $\sigma_P^2 = 4.01 \cdot 10^{-6}$.

The output of the filter for the first 80 s of the training data set is shown in Figure 6.8. The high precision of the thermistor is clearly visible, as are the offset due to self heating and the low precision of the average temperature of the pixel array. The output of the filter, using $T_0 = 22\,°\text{C}$ and $P_0 = \sigma^2$ as initialization, is shown in Figure 6.8b. The high stability of the estimated temperature is clearly visible. Using the filter outlined in this chapter the ambient temperature is estimated with accuracy of 99.9% within 5 seconds, if the initial guess has an error of $\approx 1.13\,°\text{C}$.

6.4 Object tracking

In order to track an object over time the history of its location must be collected over multiple frames. A single track is defined as the path an object follows across multiple frames. The tracking algorithm consists of five steps [33]:

- Create a distance matrix between all objects and all tracks.
- Create a correspondence matrix between all objects and all tracks.
- Assign objects to their associated tracks.
- Create new tracks for all unassigned objects.
- Remove all old tracks.

For the distance calculation three features are used.

$$\mathbf{T} = [T_x(i), T_y(i), T_A(i)] \tag{6.40}$$

where $T_x(i)$ and $T_y(i)$ represent the centroid X and Y location and $T_A(i)$ is the area of the track. The area of the track is also included as it makes the tracking more robust, by allowing only a small changes in the area of an objects between frames.

As the used features have difference magnitudes the normalized Euclidean distance measure is used to calculate the distance in 3-D feature space between each track and object. The normalized Euclidean distance differs from the Euclidean distance in that is scale-invariant. The distance between an object and a track is calculated using the following formula:

$$D(i,j) = \sqrt{\frac{(\mathbf{T}_i - \mathbf{O}_j)^2}{\mathbf{s}^2}} \tag{6.41}$$

where $\mathbf{T}_i$ and $\mathbf{O}_j$ are the feature vectors of track i and object j respectively and $\mathbf{s}$ is the covariance matrix. The covariance matrix is diagonal as the three features are assumed independent.

From the distance matrix $\mathbf{D}$ a correspondence matrix $\mathbf{C}$ is constructed [33]. This correspondence matrix contains a measure of the similarity between tracks and objects. The following three steps, outlined in [33], are used to create the correspondence matrix $\mathbf{C}$:

1. All elements of $\mathbf{C}$ are initialized to zero.
2. Find the position of the minimum element in every row $\alpha = [\alpha_1, \cdots, \alpha_m]$ and column $\beta = [\beta_1, \cdots, \beta_n]$ of $\mathbf{D}$ using the following equations:

$$D(i, \alpha_i) = \min(D(i,j)), \quad j = 1, \ldots, n \tag{6.42}$$
$$D(\beta_j, j) = \min(D(i,j)), \quad i = 1, \ldots, m \tag{6.43}$$

3. Add one to the corresponding element in matrix $\mathbf{C}$.

$$C(i, \alpha_i) = C(i, \alpha_i)) + 1, \quad j = 1, \ldots, n \tag{6.44}$$
$$C(\beta_j, j) = C(\beta_j, j) + 1, \quad i = 1, \ldots, m \tag{6.45}$$

As a result of this, each element of **C** can have one of three values: zero, one and two. A value of zero mean that neither the track nor the object have selected each other. A value of one means that either the track selected the object or visa versa and a value of two means both the track and the object have selected each other. From the correspondence matrix **C** five possible tracking situations can be obtained [33]:

- A track is not associated to any object (entire row is zero).
- An object is not associated to any track (entire column is zero).
- A track is associated to more than one object (more than one element larger than zero in a row).
- An object is associated to more than one track (more than one element larger than zero in a column).
- A track and object have a perfect match (both the row and column contain a two).

If an element in **C** contains a two, the object is assigned to the track and the corresponding cell in the matrix is set to infinite. This process is repeated until there are no more twos in the correspondence matrix. Next for all column which contains only zeros new tracks are created starting with the corresponding objects. Next all old tracks, which have a row with containing only zeros are removed.

In this resulting matrix **C** each row and column which has a one as maximum indicates a possible split or merge. However this is not guaranteed. It could, for example, also be the case that an objects moves away from this track and a similar object appears at an equal distance of the track in the opposite direction. This situation will result in a possible split event, however this is clearly not the case. As a result of this inability to accurately determine the difference between splitting, merging, creation and deletion of objects and tracks, objects are assigned to the associated track with the lowest distance in the distance matrix. Any track which is still unassigned after this step is removed. Tracks which get multiple objects assigned are duplicated. Using duplication of tracks for possible splits favours increasing the lifetime of a track as long as possible. This is advantageous as tracks with a lifetime shorter than the output filter length are not considered for output of the framework.

Chapter 7

Interaction classification module

This chapter describes the algorithms and design choices used in the classification module. The classification module consists of two distinct stages: 1) a classification stage which classifies all activities and 2) a filter stage which both limits the allowed activity transitions and reduces the output noise. The optimization of the classifiers is performed in Chapter 9.

A detailed overview of the architecture of the module is shown in Figure 7.1. It shows the classification stage highlighted in orange and the filter stage in blue respectively. In the classification stage the state of each static object is classified independently. The interactions are cross classified between all static and dynamic objects. As a result, after the classification stage, each static object has one state and n interactions associated with it, one for each

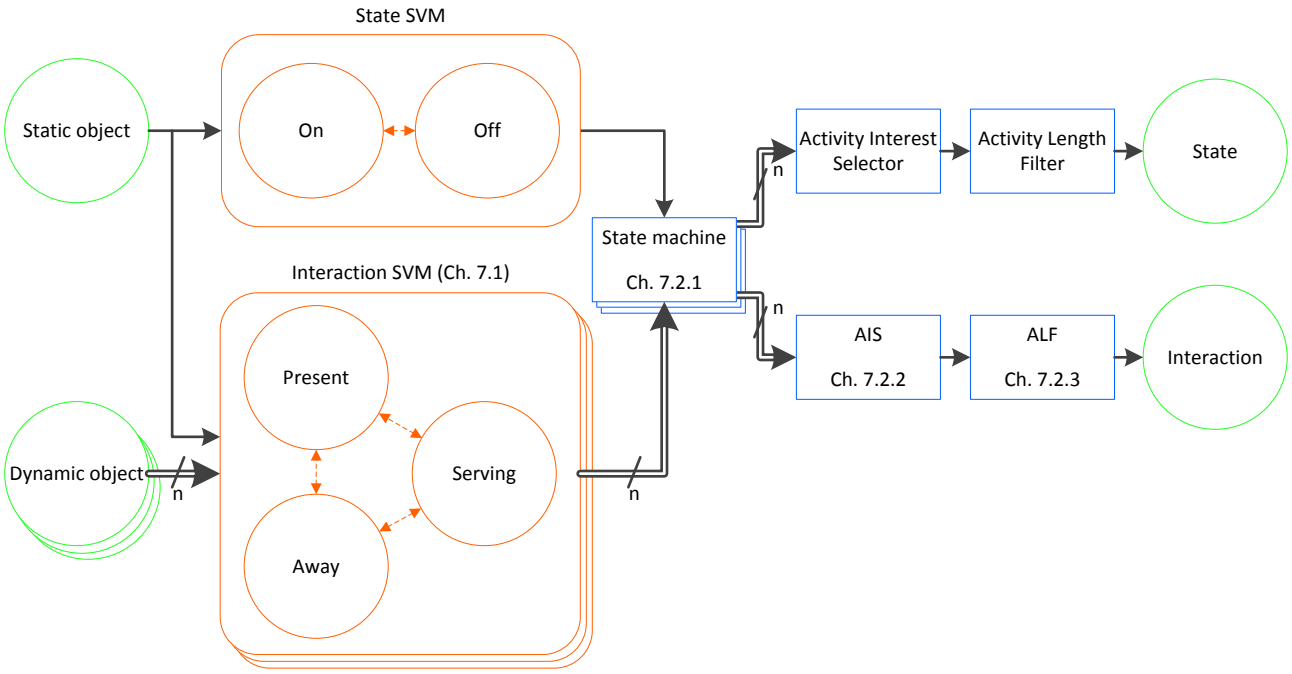


Figure 7.1: Detailed overview architecture classification module for coffee pot; shown in orange is the classification stage, shown in blue is the filter stage.

dynamic object in the scene. Next each interaction in combination with the state of the static object are passed through a state-machine. In case there is no dynamic object in the scene, the state-machine assumes a default value of "Away" for all interactions. The result of the state machine is a state limited and filtered version for each of its input. In total there are now n states and n interactions associated with a single static object. The set of n states and interactions are independently reduced to a single state and interaction using the Activity of Interest selector (AIS). As a final step the Activity Length Filter (ALF) removes all activities (states and interactions independently) from the output which have duration shorter than a threshold.

A description of the classifiers used in the classification stage is given in Section 7.1. The filter stage is described in Section 7.2.

7.1 Classifier

To determine the states and interactions for the appliances detected in the current frame a classification algorithm is used. To reduce the complexity of the classifiers, the state and the interaction for an appliance are classified independently. Furthermore the classifications of each appliance are performed independently. The total number of classifiers used in the framework is 9, two for each of the 4 appliances and one for the meeting activity.

As classification model a support vector machine (SVM) is chosen. A SVM predicts for each input vector, independently of previous input vectors, to which of the two possible classes it belongs. A basic SVM is a non-probabilistic binary linear classifier. This means a SVM classifies its input into one of only two possible classes based on a linear combination of features, without using possibly underlaying probabilities. This is applicable to the situation at hand as there is no information the underlying probability distributions. The classification parameters used by SVM to determine the decision boundary or hyper-plane used to classify the input samples are entirely obtained through training. An advantage of SVMs is that the provide good out-of-sample generalization if the parameters are chosen appropriately. Meaning, they are robust against some bias in the training set and overfitting.

Most classifications can be performed using a single SVM, as they are binary problems. However the coffee pot and refrigerator interaction, and the state of the faucet are ternary problems. As a SVM is a binary classifier, multiple SVMs are needed to solve these classification problems. For these ternary classification problems three separate SVMs are trained in a one-vs-one (OVO) fashion. The result of the classification is then obtained by combining the output of the three SVM using majority voting. OVO has as advantage over one-vs-all (OVA) that the training data set size is much smaller, reducing the time enquired to train the classifiers.

The optimal decision boundary of a SVM is defined by $\mathbf{w}^T\mathbf{x} + b = 0$. The decision boundary is considered optimal if it has a maximal margin for all training samples, meaning if the distance to all training samples is maximized. Given a training set of l training samples $\mathbf{x}_i \in \mathbb{R}^d$ labeled by class $y_i \in \{1, -1\}$, for the optimal hyper-plane the following constrain must hold [34]:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i = 1, \ldots, l \tag{7.1}$$

By minimizing $\frac{1}{2}\|\mathbf{w}\|^2$ under these constrains a unique hyper-plane is found. This method can be extended to allow for a *Soft Margin*. Soft margins can be used when Equation 7.1 has

no solution, to obtain an optimal decision boundary. Soft margins introduce slack variables which allow for misclassification of samples within a class. This results in a hyper-plane which minimizes the number of misclassification, but simultaneously maximizes the margin separating the two classes. By using soft margins the constraints from Equation 7.1 are weakened to [34]:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i = 1, \ldots, l \tag{7.2}$$

where ξ_i are the slack variables, which are constrained to $\xi_i \geq 0, i = 1, \ldots, l$ [34]. The minimization function is augmented to ensure the amount of misclassification and margin errors are as small as possible [34]:

$$\min_{\mathbf{w},\xi} \left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i \right) \tag{7.3}$$

Here C defines the relative importance of maximizing the margin and minimizing the amount of slack. The constant C will be used in Chapter 9 to optimize the classifier.

Using the method of Lagrange multipliers the dual formulation is obtained, which is expressed in terms of $\lambda_i \in [0, C], i = 1 \ldots, l$. This dual formulation has as advantage that the constrains are simpler. The resulting hyper-plane is then given by the following equation [34]:

$$g(\mathbf{x}) = \sum_{i=1}^{l} \lambda_i y_i \mathbf{x}_i^T \mathbf{x} + b \tag{7.4}$$

where b is the bias, or offset, of the classifier. The Lagrange multipliers are a notion of the influence of a training sample on the location of the decision boundary. Samples with a large and a small multiplier have a large and a small influence respectively. The Lagrange multipliers are used during the optimizations of the classifier to the find the training samples with the largest influence. The output of the SVM is then determined by the sign of $g(\mathbf{x})$. If it is negative $\mathbf{x}$ belongs to class -1, otherwise it belongs to class 1.

To increase the performance of the classifier a non-linear kernel is used. This is possible as the data samples are only used in the inner product, $\mathbf{x}_i^T\mathbf{x}$ of Equation 7.4. As kernel function a Gaussian Radial Basis Function (RBF) is used, which is defined by the following equation [34]:

$$K(u, v) = e^{-\gamma\|u-v\|^2} \tag{7.5}$$

where $\mathbf{u}$ is the hyperplane, $\mathbf{v}$ is the data sample currently being tested and γ determines the influence of a single sample, or outliers, used during training. The constant γ will be used in Chapter 9 to optimize the classifier.

The Gaussian kernel computed with a support vector is an exponentially decaying function in the input feature space [35]. The maximum of this function is located at the support vector and it decays uniformly in all directions around the support vector $\propto \gamma$. As shown in Equation 7.4, the SVM classifier with the Gaussian kernel is a weighted linear combination of the kernel function computed between a data point and each of the support vectors [35]. The role of a support vector in the classification of a data point is tempered with the Lagrange multiplier λ and $K(u, v)$, the local influence of a support vector, in prediction at a particular data point [35].

7.1.1 Feature vector

The selection of appropriate features used for classification has a great influence on the performance of the classifier. In total 18 features are identified as possibility being suitable for the classification process. Table 7.1 lists all 18 identified features. The first column in the table lists the features considered from the static object and the second column lists the feature considered from the dynamic object. The following definitions are used for the feature vector:

- Area, is the size of the object in pixels.
- Temperature, is the average temperature of the object.
- Area temperature product, is the product of the area and the average temperature of an object.
- Gradient, is the gradient around the centroid calculated with the Sharr operator 3×3. The Sharr operator has as advantage over the traditionally used Sobel operator that has perfect rotational symmetry.
- Distance, is the Euclidian distance between the centroid of two objects.
- Temperature variance, is the sample variance of temperature of pixels of an object.
- Position variance, is the sample variance of the location of the centroid of an object over the past 1 second.

Especially the area temperature product is an interesting feature. Where the area and the temperature of an object vary with the location of an object, the area temperature product remains constant. For example, an object which occupies two pixels for 50% will have a perceived temperature of only 50% of its real temperature. If the object moves slightly so it only occupies a single pixel, the pixel will have a perceived temperature of 100% of the real temperature. The area temperature product is in both cases the same, namely 1.

For the classification of interactions all 18 features are considered for the feature vector selection process in Section 9.1. For the classification of the state of an object only the static object is used, there for only 8 features are considered in the selection process.

7.2 Output filter

As all but one of the features have no temporal dependencies, the output of SVM classifiers also have no temporal dependencies. This allows for the classification of activities with a duration of only a single frame. Furthermore, as the activities are classified independently arbitrary state and interaction transitions are possible. To introduce temporal dependency and limited the allowed state transitions in the output of the classifiers three filters are used, namely:

1. A hidden Markov model (HMM) is used as state-machine to limit the allowed activity transitions, and thus remove improbable state interaction combinations.
2. An activity interest selector (AIS) is used to selected the activities which are of interest to the framework from the set of all classified activities associated with an appliance.
3. An activity length filter (ALF) is used to limited the minimum duration of an activity before it is considered for the output of the framework.

Table 7.1: Feature vector.

Static object		Dynamic object	
Index	Feature	Index	Feature
1.	Area temperature product	2.	Area temperature product
3.	Temperature variance	4.	Temperature variance
5.	Area	6.	Area
	—	7.	Distance to Static object
8.	Gradient X direction	12.	Gradient X direction
9.	Gradient Y direction	13.	Gradient Y direction
10.	Gradient magnitude	14.	Gradient magnitude
11.	Gradient phase	15.	Gradient phase
16.	Temperature	17.	Temperature
	—	18.	Position variance

7.2.1 Hidden Markov Model

In order to smooth the output of the SVMs a discrete hidden Markov model (dHMM) is used. A hidden Markov model is part of a group of models which follow the Markov property [36]. The Markov property states that the conditional probability distribution of the current state only depends on the previous state and not on the sequence of events which led to the previous state. A hidden Markov model is a special type of Markov model where the states themselves cannot directly be observed [36]. Instead the output of the states is observed, which is then used to determine the most probable state of the system for each time instance. A dHMM is modeled by two matrices, namely:

1. The transition probabilities matrix. This matrix contains the probabilities for moving to state i given current state j. This matrix is also called the Markov matrix.

2. The emission probabilities matrix. This matrix contains the probabilities of emitting an observation symbol L given the current state j.

By defining these two matrixes and specifying the initial state probabilities the model is completely defined for each time instance [36].

A HMM is chosen due to the relative simplicity at which state transitions can be modeled and limited. Also the Markov property is especially applicable to our system. Given the previous state of an appliance to most probable current state needs to be determined, for which it is safe to assume is independent of the history of the appliance before the previous state.

To increase the overall performance of the framework both the state and interactions of an appliance are used simultaneously in the HMM. This allows for the elimination of states of which it is a priori known that they do not exist. An example would be opening the refrigerator while nobody is present. Learning of the model parameters is used to automatically determine which states do not exist. For the meeting activity only the interaction classification is used, as it knows no state.

If there are no people present in the current frame, there won't be any output of the interaction SVM. In this case the HMM defaults to using the "away" interaction as default interaction.

HMMs can be used to solve three basic question about a system, namely [37]: the evaluation problem, the decoding problem and the learning problem. Of the three problems only two are of interest to the framework, namely: the learning problem, as the HMM needs to be trained to obtain the desired improve in performance, and the decoding problem, as we want to determine the most probable state given the output of the SVMs. The evaluation problem is not of interest as we are not interested in the probability of a particular sequence of observed activities.

First the general topology is given for the generated dHMM, next the implemented solutions for the learning and decoding problem are given.

Topology

For the HMM an ergodic topology is used. The used topology is the same for all appliances. By training the topology of the dHMM is shaped, by removing unused edges and rendering states unreachable, to suit the different activities. The framework uses 5 different HMMs in total, 4 for each of the appliances and one for the meeting activity.

As stated, the dHMM operators on both the state and interaction classification simultaneously. This ability is obtained by using the cartesian product of the set of states and interactions associated with an appliance. For simplicity reasons the set of state names used for HMM is equal to the set of emitted symbols. The set of state names and emission symbols, L, is defined by the following equation:

$$L = S \times I \tag{7.6}$$

where S and I are the set of states and interactions associated with an appliance respectively.

Once the most probable state inside the HMM is determined for a frame the corresponding state name is split into a separate state and interaction for the appliance. This state and interaction are then separately processed further by the activity interest selector.

Figure 7.2 shows the HMM for the coffee pot after training. As is shown, the state "Off - Serving" is removed as this state is improbable. This is due to the fact that there is no training data which contains this state. Furthermore it is shown that it is impossible to turn the coffee pot off when nobody is present. According to the training data the initial state of the HMM is either with the coffee pot "On" or "Off, but always with nobody present.

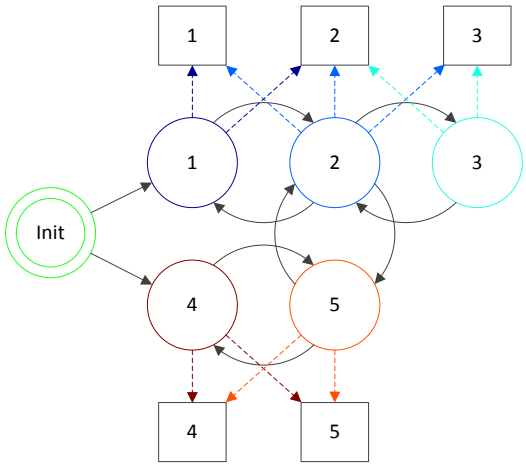
The learning problem

Using the training data set from Chapter 9, supervised training is used to train a separate dHMM for each of the 5 activities.

The method used for training is counting frequency [37]. The counting frequency algorithm consist of two steps. In the first step the number of transitions and emissions are counted in the training set. This is performed using the following equations [37]:

$$T(i, j) = \sharp \text{ of } i \rightarrow j \text{ transitions} \tag{7.7}$$

$$E(i, L) = \sharp \text{ emissions of symbol } L \text{ at state } i \tag{7.8}$$



Index	Label
1	On - Away
2	On - Present
3	On - Serving
4	Off - Away
5	Off - Present

Figure 7.2: HMM topology for coffee pot after training.

Next the transition and emission probability matrices are estimated using the respective frequencies. This is calculated using the following formulas [37]:

$$t(i,j) = \frac{T(i,j)}{\sum_k A(i,k)} \tag{7.9}$$

$$e(i,L) = \frac{E(i,L)}{\sum_s E(i,L^T)} \tag{7.10}$$

where $t(i,j)$ is the probability for a transition from state $i \rightarrow j$ and $e(i,L)$ is the probability of emitting label L in state i.

No correction is performed on the trained transition and emission probability matrices to correct for possible missing transitions/emissions in the data set.

The training method used is very simple and heavily depends on the diversity of the training set to accurate estimate emission and transition probabilities. However no optimization of the emission and transition is attempted as the main goal of the HMM is to limit the allowed activity transitions. The activity length filter allows, due to the low number of parameters compared to a HMM, for a much easier tuning of the filter.

The decoding problem

In order to determine the most probable activity for each appliance for each frame a modified version of the forward-backward algorithm is used.

The standard forward-backward algorithm uses two passes to compute the most probable state at each time instance. First the forward pass is performed. In the forward pass the probability of ending up in a particular state given all observation up to the current observation is calculated, so $P(L_{state,k}|L_{obs,1:k})$. Next the backward pass is calculated. In the backward pass the probability is calculated what is the probability of obtaining all future observation given the current state, so $P(L_{obs,k+1:t}|L_{state,k})$. In a final step the forward and backward passes are combined into a single probability for each state at every time instance. The following equation is used to combine the forward and backward pass [37]:

$$P(L_{state,k}) = P(L_{state,k}|L_{obs,1:k})P(L_{obs,k+1:t}|L_{state,k}) \tag{7.11}$$

Algorithm 7.1 iterative forward-backward algorithm.

1: **procedure** IFF($\mathbf{seq}, \mathbf{T}, \mathbf{E}, \mathbf{L}, f_s, s$)
2: $\quad \mathbf{seq} \leftarrow L(1) \cup \mathbf{seq}$ ▷ Add dummy seq to simulate previous iteration

3: $\quad$ **for** $count = 2 \rightarrow \|\mathbf{seq}\|$ **do** ▷ Forward pass
4: $\quad\quad$ **for** $state = 1 \rightarrow \|\mathbf{T}\|$ **do**
5: $\quad\quad\quad f_s(state, count) \leftarrow E(state, seq(count)) \circ \sum(f_s(:, count-1) \circ tr(:, state))$
6: $\quad\quad$ **end for**
7: $\quad\quad s(count) \leftarrow \sum f_s(:, count)$ ▷ Normalize probabilities
8: $\quad\quad f_s(:, count) \leftarrow \frac{fs(:,count)}{s(count)}$ ▷ Forward probability time instance *count*
9: $\quad$ **end for**

10: $\quad b_s(\|\mathbf{seq}\|, :) \leftarrow 1$ ▷ Initialize all final state probabilities
11: $\quad$ **for** $count = \|\mathbf{seq}\| \rightarrow 1 - 1$ **do** ▷ Backward pass
12: $\quad\quad$ **for** $state = 1 \rightarrow \|\mathbf{T}\|$ **do**
13: $\quad\quad\quad b_s(state, count) \leftarrow \sum[tr(state, :)^T \circ b_s(:, count+1) \circ E(:, seq(count))]$
14: $\quad\quad\quad b_s(state, count) \leftarrow \frac{b_s(state,count}{s(count+1)}$ ▷ Backward probability time instance *count*
15: $\quad\quad$ **end for**
16: $\quad$ **end for**

17: $\quad p \leftarrow f_s \circ b_s$ ▷ Combine forward and backward pass
18: $\quad p \leftarrow L\left\{max\left[p\left(:, \left\lceil \frac{\|seq\|}{2} \right\rceil\right)\right]\right\}$ ▷ Determine most probable state at center of seq
19: $\quad f_s \leftarrow f_s(:, 2)$ ▷ Safe forward pass for next iteration
20: $\quad s \leftarrow s(2)$
21: $\quad$ **return** $\langle p, f_s, s \rangle$
22: **end procedure**

where $L_{obs,k}$ is the label of the observed state at time instance k, $L_{state,k}$ is the label of the hidden state of the system at time instance k and t is the current time instance.

The modified version of the forward-backward algorithm used in the framework is based on [37] and the implementation of `hmmdecode`, which is part of the Matlab Statistics Toolbox. The modified version of the algorithm is presented in Algorithm 7.1. In the algorithm **seq** is the sequence of observations which is used for the decoding, f_s is the result of the forward pass of the previous iteration and s is the scaling factor used in the previous iteration. This scaling factor ensures that the probabilities can be calculated with enough precision. The main two differences with the standard algorithm in Matlab are: 1) it only return the most probable state for the time instance corresponding to the center of the sequence of observations, 2) the result of the previous forward-pass can be used for the current iteration of the algorithm. These differences are located on line 1, 18 – 21 in Algorithm 7.1.

In short the algorithm allows to calculate the most probable state at time k given all observations since the start of the recording, $L_{obs,1:t}$, while only maintaining the last $\|\mathbf{seq}\|$ observations in memory. By varying the length of **seq** the length of the backward pass can be adjusted. For the backward-pass in the framework a length of 5 is used as this yields acceptable results. No other lengths are tested.

Table 7.2: Activity interest level ranking; S = state and I = interaction.

Rank	Coffee pot		Faucet		Refrigerator		Microwave		Meeting	
	S	I	S	I	S	I	S	I	S	I
1	On	Serving	Hot	Present	Open	Present	On	Present	—	Yes
2	Off	Present	Cold	Away	Closed	Away	Off	Away		No
3		Away	Off							

7.2.2 Activity of interest selector

At this point each appliance has n activates, associated with it, one for each dynamic object. It is assumed that each appliance only supports single-user activities. As a result the activity of interest selector (AIS) is used to select the most interesting activity which is currently being performed by the appliance. The interest level ranking for each activity is specified in Table 7.2. The interest level of an activity is loosely defined as the amount of information the framework can extract from an activity. As the framework is designed to extract information concerning energy consumption from a scene, activities are ranked by their energy consumption. Alternatively, it can be stated that the activities are ranked according the potential of energy saving they possess. Activities which have the potential to allow for large energy savings are ranked first and activities which have no or a very low potential are ranked last.

The AIS uses the following two equations the determine the most interesting activity amongst the n currently performed by the appliance.

$$\mathbf{A} = \{a \in \mathbf{D} : \forall b \in \mathbf{D}\,[\text{Interest}(a) \leq \text{Interest}(b)]\} \tag{7.12}$$

$$\mathbf{O} = \bigcup_{a \in \mathbf{A}} a \tag{7.13}$$

Here $\mathbf{D}$ is the set of all interactions associated with an appliance in the current frame, $\mathbf{A}$ is the set of the activities with the highest interest level currently performed by the appliance and $\mathbf{O}$ is the resulting activity assigned to the particular appliance for the current frame. Interest(x) returns the interest rank of activity x and a and b are two activities from the set $\mathbf{D}$. As all interest levels are unique, all activities in set A are, by definition, the same. The union of $\mathbf{A}$, $\mathbf{O}$, contains therefore by definition only a single activity.

To keep track of the classified activities per object over time the AIS creates two sets, $\mathbf{I}_s$ and $\mathbf{I}_i$, per object. One set contains the classified, and HMM filtered, states of the object and one contains the classified interactions. The sets are buildup of triplets. Each triplet consists of the activity name, the start frame index and the end frame index. The triplets are stored in chronological order inside the set, where the last item corresponds to the classifications

performed for the current frame. This is expressed by the following equations:

$$\mathbf{I}_s(k) = \begin{pmatrix} \langle A^1_{state}, 1, t_{s,1} \rangle \\ \vdots \\ \langle A^N_{state}, t_{s,N}, N \rangle \end{pmatrix} \tag{7.14}$$

$$\mathbf{I}_i(k) = \begin{pmatrix} \langle A^1_{interaction}, 1, t_{i,1} \rangle \\ \vdots \\ \langle A^N_{interaction}, t_{i,N}, N \rangle \end{pmatrix} \tag{7.15}$$

where k is the index of the object, A is the classified activity, $t_{x,1}$ and tx, N are the end time and start time of the first and last classified activity respectively and N is the current frame index.

7.2.3 Activity length filter

The output of the AIS may still contain activities with a duration of only a single frame. These short activities are of no interest, as none of the selected activates can be performed in such a sort time frame. The activity length filter (ALF) is used to remove all activities from the output of the AIS which have a duration shorted than a specific threshold τ. The optimal value of τ is determined in Chapter 10.

The ALF is effectively a discrete low-pass filter. The algorithm used for the ALF is outlined in Algorithm 7.2, here $\mathbf{I}$ is either the set $\mathbf{I}_s$ or $\mathbf{I}_i$. I^1_{start} and I^N_{end} are the second and the third item from the first and the last triplet form list $\mathbf{I}$. a_{act}, a_{start} and a_{end} are the first second and third item form the triplet respectively and a^{+1} is the next triplet from the set $\mathbf{I}$.

The algorithm starts by removing all activities with a duration shorter than the window length τ. Next all activities which are preceded by the same activity are removed. This ensures that an activity is never followed by the same activity. Finally, all activities are realigned by setting the end time of an activity to the start time of the next activity minus one frame. To make sure that the total duration of filtered set of activities is equal to that of the unfiltered set, the start and end times are overwritten using those of the unfiltered set.

It is possible for the algorithm to return an empty set. This is the case when the object has not performed any activity for longer length than the filter window. In this case the activity with the lowest ranking, according to Table 7.2, is assigned to object.

In the current implementation of the framework the two sets used in the AIS, $\mathbf{I}_s$ and $\mathbf{I}_i$, are each iteration overwritten with their respected filtered versions obtained by passing them though the ALF. This however is not required. It is enough to overwrite them with only the last triplet from the filtered set. Instinctively, this can be reasoned from the fact that only the last triplet in the $\mathbf{I}_s$ and $\mathbf{I}_i$ set can change as a result of the AIS. As only the last item of the input set of the ALF can change, also only the last item in the output set can change.

Algorithm 7.2 Activity length filter algorithm (ALF).

1: **procedure** ALF($\mathbf{I}$)
2: $start \leftarrow \min(\mathbf{I}_{start})$ ▷ Get start first activity
3: $end \leftarrow \max(\mathbf{I}_{end})$ ▷ Get end last activity
4: $\mathbf{I} \leftarrow \{a \in \mathbf{I} : a_{end} - a_{start} < \tau\}$ ▷ Remove all activities shorter than τ
5: $\mathbf{I} \leftarrow \left\{a \in \mathbf{I} : a_{act} \neq a_{act}^{+1}\right\}$
6: **if** $\|\mathbf{I}\| > 0$ **then** ▷ Check if I is not the empty set
7: **for all** $a \in \mathbf{I}$ **do** ▷ Realign activities
8: $a_{end} \leftarrow a_{start}^{+1} - 1$
9: **end for**
10: $I_{start}^{1} \leftarrow start$
11: $I_{end}^{N} \leftarrow end$
12: **end if**
13: **return I**
14: **end procedure**

Chapter 8

Experimentation

This chapter gives a description of the experiments which were performed to obtain the training and validation data set.

8.1 Test setup

The data sets are recorded in the pantry of floor 3, in the Potentiaal building at the TU/e campus. For the recordings an evaluation kit of the Panasonic Grid-EYE sensor [24] is used. The evaluation kit consists of a Grid-EYE sensor (AMG8851, high gain version) connected to Arduino Duemilanove development board. The Arduino uses the Atmel ATmega328 as its processing unit. The Arduino is programmed to query the sensor at 10 Hz and output the raw pixel values in a comma-separated values (CSV) format over the debug output. The debug output is saved to a file to be processed later. The complete development kit from Panasonic is shown in Figure 8.1, the Grid-EYE sensor is located in the center left part of the image.

8.1.1 Sensor mounting

The mounting location of the sensor is of great importance for correct classification of the actions. Due to the limitations described in Chapter 4.1.1 it is important that the sensor has an unobstructed view of all appliances. As the faucet, coffee pot and refrigerator all involve activities which are performed in front of the user, the user cannot be located in between the sensor and the appliance. As these three appliances are placed against a wall it is undesirable

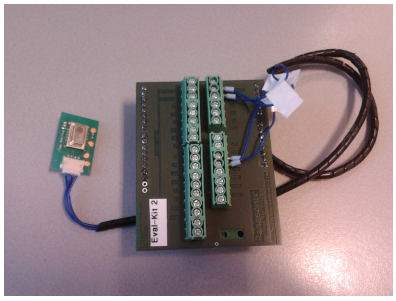
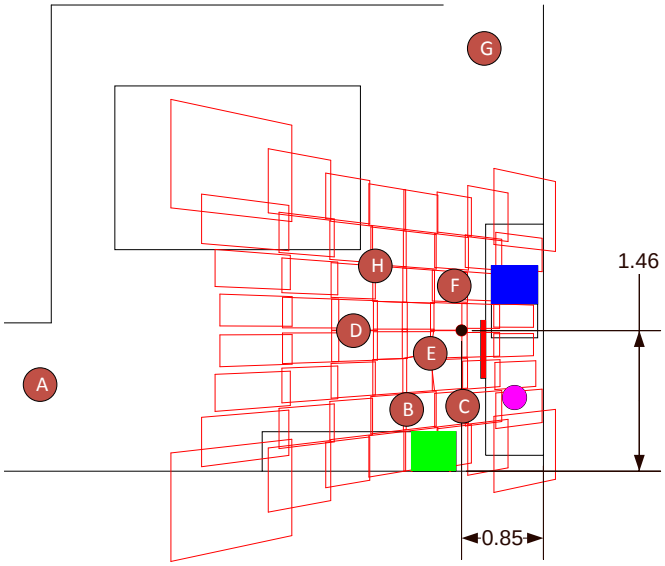


Figure 8.1: Panasonic Grid-EYE development kit.



Legend	
A	Enter / exit point 1
B / green	Microwave
C / magenta	Coffee pot
D	Halfway point A-G
E / red	Refrigerator
F / blue	Faucet
G	Enter / exit point 2
H	Meeting point

Figure 8.2: Thermopile matrix projection, the sensor is mounted under a polar angle of 15 °.

to place the sensor right above these appliances and have it looking straight down. This would result in half of the frame being occupied by the wall. Instead the sensor is placed under a polar angle of 15 °. The resulting pixel array projection is shown in Figure 8.2. As shown in the figure, the two walls are aligned to the right and bottom side of the frame allowing for optimal use of the FOV of the sensor.

It is shown in Figure 8.2 that the area of the pixel projections differ greatly between the pixels on the left side and right side of the image. For this reason the area compensation is performed during lens correction.

8.2 Data set description

The recorded data set consists of two parts. One scripted part which is used for training the classifiers and one unscripted part used to evaluate the performance of the framework. The two data sets are recorded on two separated days. The sensor has been repositioned between the training set recording and the validation set recording. As a result the alignment of the objects inside the scene differs slightly between the two set. This difference will aid in determining the robustness of the classification algorithms. The difference between the two scenes is shown in Figure 8.3. From the figures it can be concluded that the ambient temperature was 1 K higher compared to the training set. Also the coffee pot has been moved between the two data set, this has probably happened when somebody made a fresh pot of coffee. Thirdly the validation recording is rotated 180 ° compared to the training recording. To compensate for this the validation recording is rotated 180 ° before it is passed trough the framework.

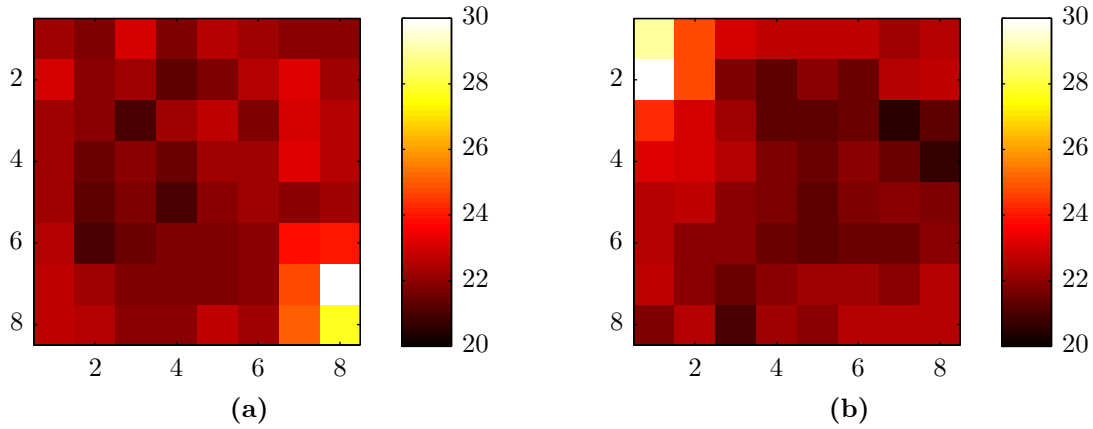


Figure 8.3: Differences between training and validation data set; (a) training data set, (b) validation data set.

8.2.1 Training data set

The training data set consist of 21 scripted scenarios and one control scenario which contains no activity. The list of all 22 scenarios is given in Table 8.1, the markers used in the table correspond to the markers in Figure 8.2. The table gives for each training set the type of activity performed, the number of times the activity is repeated and the total duration of the training set. The total duration of the entire training set is 31 minutes and 41 seconds.

Tables 8.2, 8.3, 8.4, 8.5 and 8.6 list the total number of isolated activity patterns and the total training length for each activity.

The ground truth (GT) recoding for the training data set is made by hand on a frame-by-frame basis. As a result the state and interactions for all appliances are known for each frame in the training set.

8.2.2 Validation data set

The validation data set consists for 5 hours of unscripted recording (177772 frames @ 10 Hz). The number of times each activity is performed and the total duration in the validation data set are listed in Table 8.7. During the recoding of the data set one additional appliance was used, which is not present in the training data set, namely an electric kettle. This kettle was placed next to the coffee pot. Interactions performed with the kettle are annotated in the GT as if they where performed with the coffee pot.

The ground truth recoding for the data set is performed using a video recorded at 1 FPS. Two consecutive frames from the video are shown in Figure 8.4. The angle at which the video camera has been placed has as disadvantage that it is difficult to distinguish between a person standing in front of the faucet and actually using the faucet. A second disadvantage is that it is impossible to distinguish if somebody is using hot or cold water. As the sample rate of the ground truth is only $^{1}/_{10}{}^{\text{th}}$ that of the data set recording it is up-sampled by repeating each GT annotation 10×.

8.3 Evaluation metrics

The performance of the classification framework is measured using precision and recall. Due to the annotation differences different states and interactions are evaluated for the training and validation set. Table 8.8 lists the activities which are in used in the performance evaluation for each of the two sets. The precision for a single state or interaction is calculated using the

Table 8.1: Training data set activity patterns.

Data set	Action			♯	Duration
-	Control measurement,	all appliances off		1	1:54
1	Single person,	$A \rightarrowtail$ D	$\rightarrowtail G$	3	0:56
2	Single person,	$A \rightarrowtail$ D	$\rightarrowtail A$	2	0:50
3	Single person,	$A \rightarrowtail$ D (wait, 10 s)	$\rightarrowtail G$	3	1:18
4	Single person,	$A \rightarrowtail$ D (wait, 10 s)	$\rightarrowtail A$	3	1:09
5	Single person,	$A \rightarrowtail$ C (use, 1 serving)	$\rightarrowtail A$	3	2:17
6	Single person,	$G \rightarrowtail$ F (use cold, 10 s)	$\rightarrowtail A$	3	1:26
7	Single person,	$A \rightarrowtail$ F (use hot, 10 s)	$\rightarrowtail G$	4	2:01
8	Single person,	$G \rightarrowtail$ B (use, 1 min)	$\rightarrowtail G$	1	1:36
9	Single person,	$G \rightarrowtail$ B (use, 1 min)	$\rightarrowtail G$	1	1:32
10	Single person,	$G \rightarrowtail$ B (use, 1 min)	$\rightarrowtail G$	1	1:29
11	Single person,	$G \rightarrowtail$ E (use, 5 s)	$\rightarrowtail G$	3	1:13
12	Two persons (2 s delay),	$A \rightarrowtail$ D	$\rightarrowtail G$	4	1:29
		$A \rightarrowtail$ D	$\rightarrowtail G$		
13	Two persons (2 s delay),	$A \rightarrowtail$ D	$\rightarrowtail G$	3	1:06
		$A \rightarrowtail$ D	$\rightarrowtail A$		
14	Two persons (2 s delay),	$A \rightarrowtail$ D	$\rightarrowtail G$	3	1:14
		$A \rightarrowtail$ D	$\rightarrowtail G$		
15	Two persons (2 s delay),	$A \rightarrowtail$ D	$\rightarrowtail G$	3	1:28
		$A \rightarrowtail$ D	$\rightarrowtail A$		
16	Two persons (5 s delay),	$G \rightarrowtail$ C (use, 1 serving)	$\rightarrowtail A$	3	1:47
		$A \rightarrowtail$ D	$\rightarrowtail G$		
17	Two persons (3 s delay),	$G \rightarrowtail$ E (use, 5 s)	$\rightarrowtail G$	3	1:21
		$A \rightarrowtail$ D	$\rightarrowtail G$		
18	Two persons (5 s delay),	$G \rightarrowtail$ F (use cold, 10 s)	$\rightarrowtail G$	3	1:31
		$G \rightarrowtail$ E (use, 5 s)	$\rightarrowtail A$		
19	Two persons (30 s delay),	$G \rightarrowtail$ B (use, 1 min)	$\rightarrowtail G$	1	1:29
		$A \rightarrowtail$ C (use, 1 serving)	$\rightarrowtail A$		
20	Two persons (30 s delay),	$G \rightarrowtail$ B (use, 1 min)	$\rightarrowtail G$	1	1:28
		$A \rightarrowtail$ C (use, 1 serving)	$\rightarrowtail A$		
21	Two persons (Simultaneously),	$A \rightarrowtail$ H (wait, 10 s)	$\rightarrowtail A$	3	1:18
		$G \rightarrowtail$ H (wait, 10 s)	$\rightarrowtail G$		

Table 8.2: Training data set coffee pot.

Action		♯ isolated activity patterns	Total pattern length
State	Off	1	0:38
	On	3	2:10
Interaction	Away	15	0:39
	Present	17	1:09
	Serving	7	1:00

Table 8.3: Training data set faucet.

Action		♯ isolated activity patterns	Total pattern length
State	Off	8	0:52
	Hot	3	0:43
	Cold	3	0:26
Interaction	Away	8	1:31
	Present	6	0:31

following formula:

$$\text{precision} = \frac{|\text{retrieved}|}{|\text{retrieved} \cup \text{insertions}|} \tag{8.1}$$

$$\text{recall} = \frac{|\text{retrieved}|}{|\text{retrieved} \cup \text{deletions}|} \tag{8.2}$$

Table 8.4: Training data set microwave.

Action		♯ isolated activity patterns	Total pattern length
State	Off	6	0:23
	On	3	3:39
Interaction	Away	9	3:17
	Present	6	0:45

Table 8.5: Training data set refrigerator.

Action		♯ isolated activity patterns	Total pattern length
State	Closed	8	0:19
	Open	6	1:07
Interaction	Away	8	0:19
	Present	6	0:48
	Interacting	12	0:20

Table 8.6: Training data set meeting.

Action		♯ isolated activity patterns	Total pattern length
Interaction	No	34	8:12
	Yes	8	0:45

(a) **(b)**

Figure 8.4: Two consecutive frames from the validation data set annotation; (a) shows coffee being served, (b) shows away for all appliances.

where retrieved is the set of recognized activities, insertions is the set of recognized activities which are not present in the ground truth and deletions is the set of activities which are present in the ground truth but are not recognized. Besides the precision and recall of individual states and interactions, the overall performance of an appliance is also calculated. The overall performance is calculated used the following equation:

$$\text{precision}_I = \frac{\sum_{i \in I} |\text{retrieved}_i|}{\sum_{i \in I} |\left(\text{retrieved}_i \cup \text{insertions}_i\right)|} \tag{8.3}$$

$$\text{recall}_I = \frac{\sum_{i \in I} |\text{retrieved}_i|}{\sum_{i \in I} |\left(\text{retrieved}_i \cup \text{deletions}_i\right)|} \tag{8.4}$$

where I is the set of all states or interactions for a specific appliance.

Table 8.7: Validation data set activity patterns.

Appliance	Action		♯ isolated activity patterns	Total pattern length
Coffee pot	State	Off	0	—
		On	1	4:56:17
	Interaction	Serving	27	0:04:32
Faucet	State	Off	39	4:48:58
		Hot / Cold	38	0:07:19
	Interaction	—	—	—
Microwave	State	Off	10	4:31:05
		On	9	0:25:12
	Interaction	—	—	—
Refrigerator	State	Closed	7	4:55:12
		Open	6	0:01:05
	Interacting	Interacting	12	0:00:25
Meeting	Interaction	No	24	4:43:30
		Yes	23	0:12:47

Table 8.8: Evaluated activities for training and validation data set.

Appliance	Training set		Validation set	
	State	Interaction	State	Interaction
Coffee pot	On Off	Away Present Serving	On Off	Serving
Faucet	Off Hot Cold	Away Present	Off Hot / Cold	—
Microwave	On Off	Away Present	On Off	—
Refrigerator	Open Closed	Away Present Interacting	Open Closed	Interaction
Meeting	—	Yes No	—	Yes No

Chapter 9

Classification module tune-up

This chapter gives a detailed overview of the procedures used to train and subsequently tune the support vector machines used for classification of the different activities. All results presented in this chapter are obtained by using the individual samples from the isolated activity patterns from the training data set. Object detection, tracking and the output filter are not used. Due to the very large size of the isolated activity patterns a preprocessing step was preformed to select the samples which are used in the feature vector selection process. Only a subset of all activity patterns is used for the selection process to speed up the computation time. During training of the final optimal classifier all training samples are used. This optimal classifier is then used to obtain the results in Chapter 10.

The preprocessing step limits the maximum number of samples used during training of the classifier to 1000 samples. The limit of 1000 samples is chosen as the optimal tradeoff between computation time and training set size. With this limit the complete selection and training process of all classifiers takes around 10 hours. To select the subset of samples used during the feature vector selection process and the classifier tuning process the following procedure is used:

1. Create a set of samples for each of the activities which is going to be classified.
2. Fill the sets with all samples from the training set.
3. If the union of all sets contains more than 1000; remove one sample at random from the largest set.
4. Repeat step 3 until the union of all sets contains at most 1000 samples.

This selection process aims to create equally sized sets for each of the activities to be classified. This is desirable over maintaining the size ratio between the sets, as for training of the classifier it is desirable for each class to have as many samples as possible. The precise number of samples used during selection and optimization process for each activity is listed in Table 9.1.

9.1 Feature vector reduction

To reduce the complexity of a classifier it is desirable to use as few features possible, while maintaining an acceptable performance level. The procedure outlined in [34] is used to create an optimal subset of the features. The main idea presented in [34] is to obtain a feature

Table 9.1: Feature vector selection training set size.

Appliance	State			Interaction		
	Off	On		Away	Present	Serving
Coffee pot	379	621		333	333	334
	Off	Hot	Cold	Away	Present	
Faucet	370	370	260	695	305	
	Off	On		Away	Present	
Microwave	226	774		554	446	
	Closed	Open		Away	Present	Interacting
Refrigerator	185	674		185	475	199
				No	Yes	
Meeting				549	451	

importance ranking by first train the classifier using the default tuning parameters, $\gamma = 1$ and $C = 1$, and all available features. Using this ranking an optimal subset of features is determined and the γ and C parameters are tuned to obtain the maximum performance level. The disadvantage of this technique is that it assumes that all features are independent, which is clearly not the case for the used feature vector. For example, feature 8 and 9 (gradient X and Y direction) have some correlation with feature 10 and 11 (gradient magnitude and phase). A similar case is applicable to feature 1, 5 and 16 (area temperature product, area and temperature). From experiments it is concluded that using the entire feature vector for the initial training results, due to this high degree of correlation, in a very poor overall performance. As a result ranking of the features would be far from optimal, which results in the selection of an unnecessary large feature vector. Instead of training on the entire feature vector, the classifier is first trained on each feature independently. This individual training eliminates the influence of inter-feature dependency on the estimated feature importance. As a result, features which have a correlation will have a similar performance. A downside of this however is that as these correlated features contain to some extent the same information, when combined the overall performance of the classifier will only marginally increase, or even decrease. Therefore it is still required to perform an additional manual step at the end of the selection process, where high performance feature which don't contribute to the performance of the classifier are removed from the feature vector.

The training on individual features results in an optimal set of Langrange multipliers λ_i per feature, which in turn shape the support vectors and decision boundary. To rank the features in the feature vector according to their influence on the classification decision, the gradient of hyper-plane is calculated. The local gradient of the hyper-plane is calculated using

the weighted sum over all support vectors using the derivative of the kernel function [34]:

$$g(\mathbf{x}) = \sum_{i=1}^{l} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{9.1}$$

$$\nabla g(\mathbf{x}) = \sum_{i=1}^{l} \lambda_i y_i \nabla_x K(\mathbf{x}_i, \mathbf{x}) \tag{9.2}$$

where $g(\mathbf{x})$is the separating hyper-plane, λ_i are the Langrange multipliers, y_i are the training labels ($\mathrm{y}_i \in \{1, -1\}$), $K(\mathbf{x}_i, \mathbf{x})$ is the kernel function and b is the bias. The derivative of the RBF kernel function is given by the following equation:

$$\nabla_x K(\mathbf{x}_i, \mathbf{x}) = 2\gamma(\mathbf{x}_i - \mathbf{x})\mathrm{e}^{(-\gamma\|\mathbf{x}_i - \mathbf{x}\|^2)} \tag{9.3}$$

After normalization $\nabla g(\mathbf{x})$ is compared to the unit vector $\mathbf{e}_j$, $j = 0, \ldots, n$, representing the indices of the different features. If feature $\mathbf{x}_j$ contributes no information to the classification at position $\mathbf{x}$, $\nabla g(\mathbf{x})$ should be orthogonal to $\mathbf{e}_j$. The angle between the unit vector $\mathbf{e}_j$ and $\nabla g(\mathbf{x})$ is calculated using the following equation [34]:

$$\alpha_j(i) = \min_{\beta \in \{0,1\}} \left[\beta\pi + (-1)^\beta \arccos\left(\frac{(\nabla g(\mathbf{x}_i))^T \mathbf{e}_j}{\|\nabla g(\mathbf{x}_i)\|} \right) \right] \tag{9.4}$$

where β is used to make α_j sign insensitive, meaning independent of whether the local gradient is positive or negative. $\alpha_j \approx \pi$ indicates that the respective feature only has very limited influence on the location of the hyper-plane, whereas $\alpha_j \approx 0$ indicates that it has extensive influence [34].

$\mathbf{x}_i$ located far away from the discission boundary have only limited influence on its location, independent of the projections onto the unit vector. Therefore α_j should only be evaluated for $\mathbf{x}$ located close to the decision boundary, however $\mathbf{x}_i$ located on the wrong side of the boundary should not be evaluated as they are incorrectly classified.

In [34] $\mathbf{I}_\in = \{\mathbf{x}_i \in \mathbf{x} : |g(\mathbf{x}_i) - 1| \leq \epsilon\}$ is used to define which points are included in the performance evaluation. A downside of this selection criteria is that it requires a tuning parameter ϵ. If ϵ is too small not enough points are included and the result will be highly susceptible to noise, if ϵ is too large too many points are included and the points far from the decision boundary will negatively influence the result. This is because they contribute equally to the ranking of the features, but they do not influence the decision hyper-plane.

Instead a new selection criteria is proposed in this work, which selects the 20 most influential points of each class. The influence is measured using the Langrange multipliers, the larger the multiplier the greater the influence a point has on the decision boundary. The advantage over the original metric is that the new metric is gradient independent. The value of 20 has proven to yield good results, however for truly optimal results this number should be optimized for each classifier. This step is not performed and for all results presented in this thesis a total set size of 40 points is used. The selection criteria is formally defined as:

$$\mathbf{I}_\in = \bigcup_{c=\{1,-1\}} \left\{ s \in \mathcal{P}(\lambda)_c : \|s\| = 20 \wedge \forall_{j \in \mathcal{P}(\lambda)_c} \left(\sum s \geq \sum j \right) \right\} \tag{9.5}$$

where $\mathcal{P}(\lambda)_c$ is the power set of all Langrange multipliers corresponding to points of class c and $\mathbf{I}_\in$ is the set of points used for the performance evaluation.

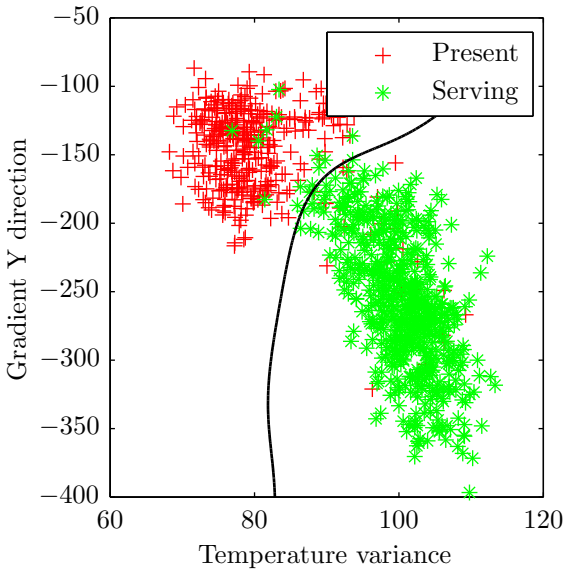


Figure 9.1: Example from the coffee pot data set showing a $\widetilde{\alpha} \approx 0.5$ for both the temperature variance and gradient y-direction.

The resulting angles of Equation 9.4 using set $\mathbf{I}_{\in}$ are averaged and scaled to a single angle per feature using the following equation [34]:

$$\widetilde{\alpha}_j = 1 - \frac{2}{\pi} \cdot \frac{\sum_{i \in I_{\in}} \alpha_j(i)}{|I_{\in}|} \tag{9.6}$$

where $\widetilde{\alpha}_j = 1$ and $\widetilde{\alpha}_j = 0$ indicate a strong and weak influence respectively. Given $\widetilde{\alpha}$ a ranking is established amongst the different features. The feature with the largest $\widetilde{\alpha}$ is ranked first. The feature with the smallest $\widetilde{\alpha}$ is ranked last. Using this ranking the feature vector is created by continuously adding features, starting with the feature with the highest ranking, until a satisfactory performance is obtained.

For a better perception of the meaning of the magnitude of $\widetilde{\alpha}$ Figure 9.1 is added. In this figure a scatter plot shows the temperature variance and y-direction gradient from the coffee pot training set. As the decision hyper-plane is nearly parallel with the line $y = x$, it indicates that both the variance and the y-direction gradient contain a similar amount of information.

Figures 9.2 to 9.6 show the performance of the individual features calculated using Equation 9.6. The figures show two groupings of the data. Grouping 1 shows the feature importance for the state classifier and grouping 2 for the interaction classifier. The labels on the x-axis correspond with the index for the feature in the feature vector, see Table 7.1 for indices. For classifiers using mutli-class SVM, $\widetilde{\alpha}$ is averaged across the individual binary classifiers [34].

The feature performance for the coffee pot is shown in Figure 9.2. Grouping 1, the state classifier, clearly shows two features of high importance, the temperature variance and gradient in the Y direction. The average temperature ranks in fifth, after the gradient in the X direction and the gradient phase. Closely followed by the gradient magnitude. This clearly shows that feature which have a high correlation are also closely ranked. For grouping 2, the interaction classifier, the top ranking features are temperature and temperature variance of the person. It is peculiar, but explainable, that the distance and position variance have a low

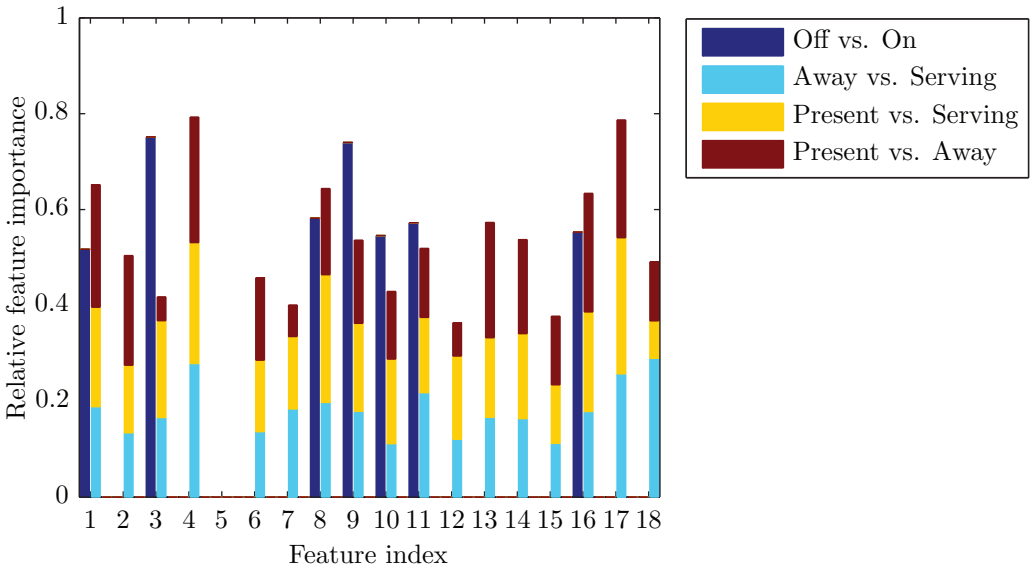


Figure 9.2: Feature performance coffee pot classifier; grouping 1 is the state classifier, grouping 2 is the interaction classifier.

ranking. For the distance feature it is shown that even for the "away vs. serving" classifier it has a low ranking. This is due to ability of being close to the coffee machine in the "off" state. It is shown in Figure 9.2 that the position variance feature ranks as the most important feature to classify between "away" and "serving", however for "present vs. serving" and "present vs. away" it provides only very limited information. This is explainable as in both the "away" and "present" interaction the position variance is large, whereas in the "serving" interaction it is small as the person is stationary. Feature 5 contains no information as the area of the coffee pot is set to a fixed size. Also the position variance, feature 18, contains no information as the coffee pot is always stationary. It is the case for state classifiers for all appliances that feature 5 and 18 contain no information.

The feature performance for the faucet is shown in Figure 9.3. When compared to the coffee pot state classifier it is clear that this classifier will perform worse as all features combined contain about 20% less information than all features combined for the coffee pot. The top ranking feature of the faucet contains only 62.5% of the information the top ranking feature of the coffee pot contains. The top ranking feature for the state classifier is the temperature. This is mainly due to the "off vs. hot" classifier. The temperature variance is the top ranking feature for the "off vs. cold" classifier, however with only 40.8% of the maximum amount of information the classification will still be very poor.

The feature performance for the refrigerator is shown in Figure 9.4. The top ranking features for the state classifier are the temperature area product and the temperature of the refrigerator. As the refrigerator is a static object its area is constant. As a result the temperature area product is a scaled version of the temperature feature. Using both features would thus be pointless. For the interaction classifier the temperature variance is the top ranking feature. The large temperature variance is caused by the person browsing through the refrigerator during the "interacting" interaction.

The feature performance for the microwave is shown in Figure 9.5. Similar to the faucet

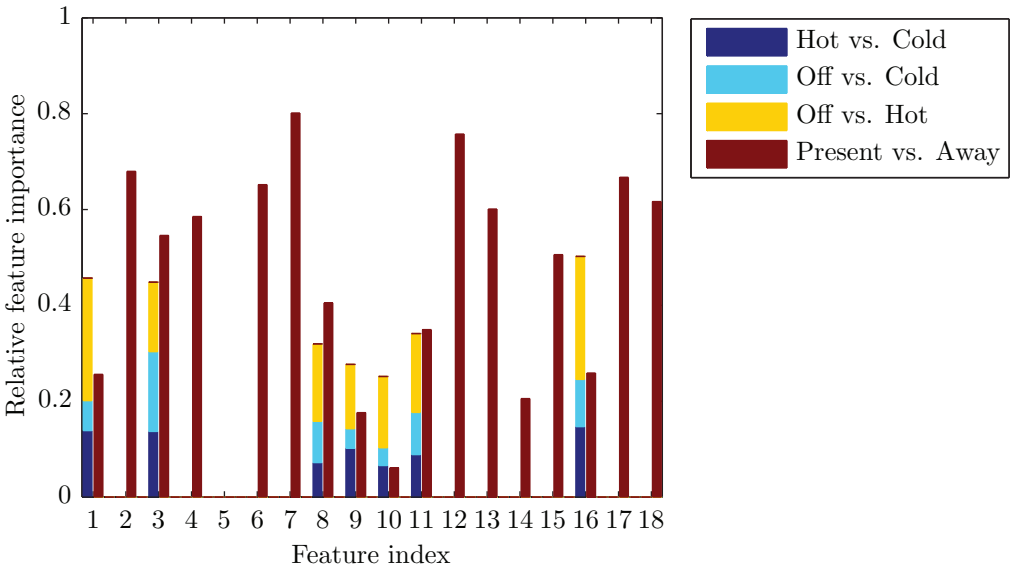


Figure 9.3: Feature performance faucet classifier; grouping 1 is the state classifier, grouping 2 is the interaction classifier.

the state of the microwave is difficult to classify, as the visible differences between the "on" and the "off" state are only very small. A microwave which is turned on doesn't heat up much and once it has heated up it will remain hot for a long period even when turned off. The top ranking feature is the gradient, which can be explained by the uneven heating of the microwave. The housing covering the active area becomes warmer than the part covering

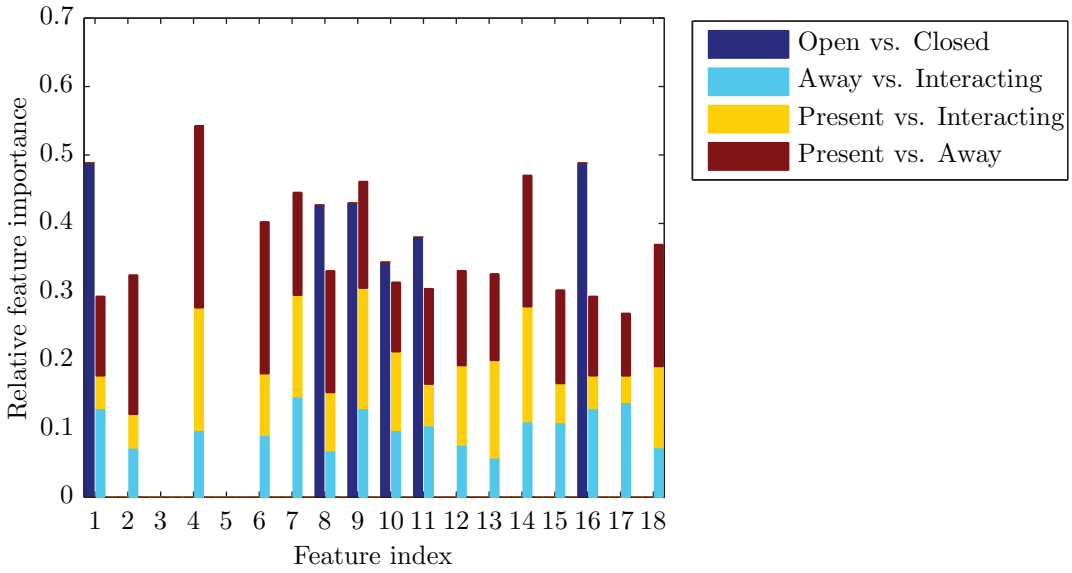


Figure 9.4: Feature performance refrigerator; grouping 1 is the state classifier, grouping 2 is the interaction classifier.

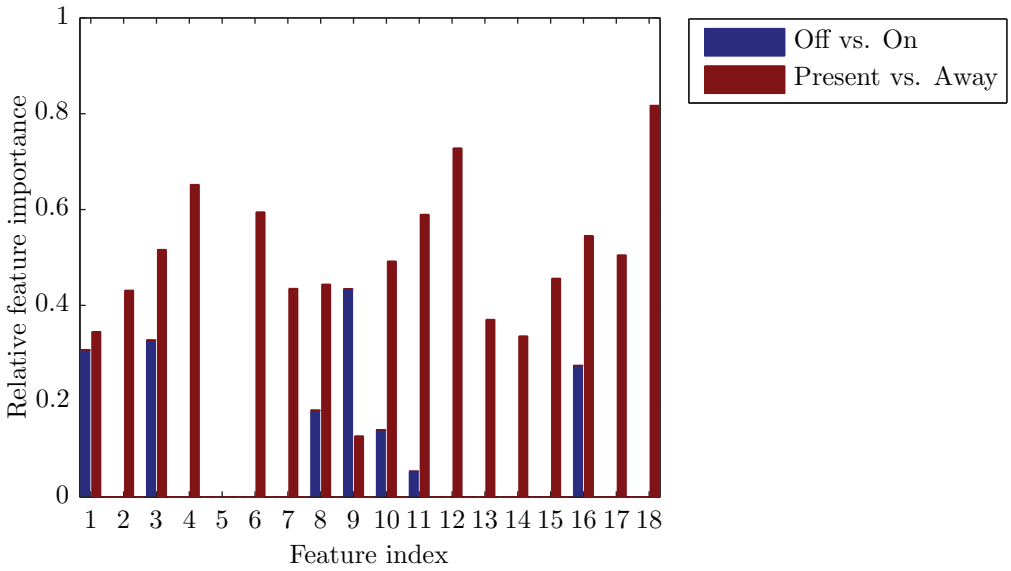


Figure 9.5: Feature performance microwave; grouping 1 is the state classifier, grouping 2 is the interaction classifier.

the electronics. For the interaction classification the top ranking features are the position variance and temperature gradient in the X direction of the person. The importance of the position variance can be an artifact of the training data set as during the training the persons stood stationary the entire length of time it took to prepare the food. In the validation data set it has been observed that this not natural behaviour.

The feature performance for the meeting is shown in Figure 9.6. The top ranking features are the temperature variance and gradient magnitude, and area. This is caused as the resulting object of two persons standing next to each other has a clear gradient. The phase of the gradient has little information as the two persons can in any orientation stand side by side.

Using the generated ranking the performance of each classifier is determined to find the optimal combination of features for classifier.

9.2 Classifier performance

Using the previously determined ranking of the features, the optimal combination of features for each classifier is determined. Similar to [34], the optimal combination is found by successively adding features to the feature vector until a satisfactory performance is obtained. Accuracy is used as the metric to determine the performance of the classifier. The definition for accuracy used is:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{9.7}$$

where TP and TP are the number of true positives and true negatives and FP and TN are the number of false positives and false negatives.

The accuracy of the state classifiers is shown in Figure 9.7 and in Figure 9.8 for the interaction classifiers. Coloured in glyphs indicate the selected features based on their rank.

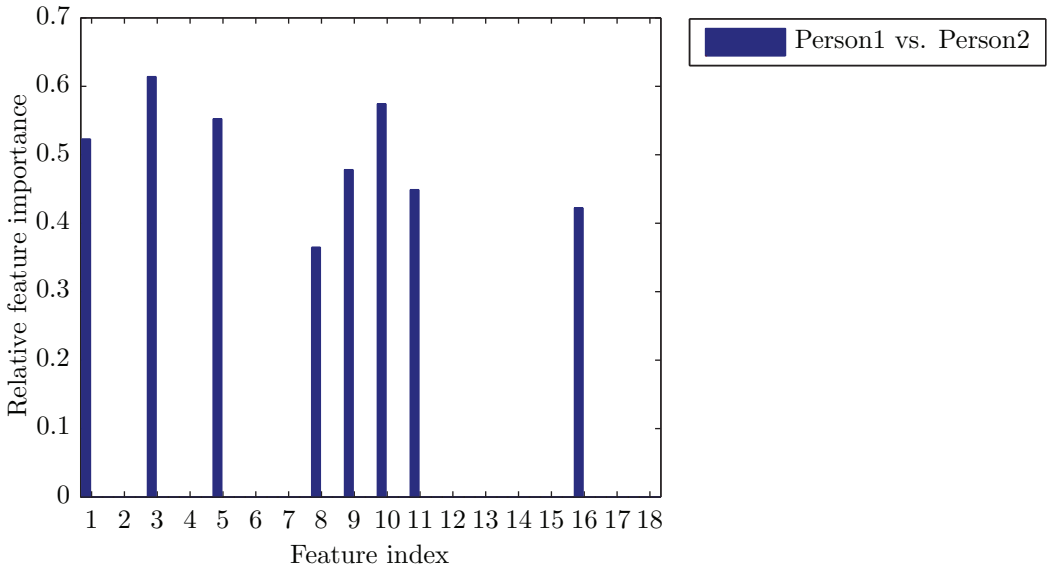


Figure 9.6: Feature performance meeting classifier.

The accuracy of the classifiers is determined using k-fold cross validation on the training data set. The number of features selected for each classifier is based on a combination of the accuracy obtained in relation to the maximum accuracy and the number of selected features.

From Figure 9.7 it can be concluded that by adding a second feature, no increase in performance is obtained. This because the first feature is the temperature and the second feature is the area-temperature product. As the area is constant no information is added by the second feature. It is shown that for both the faucet and the microwave a large number of features are required in order to obtain an expectable performance level. This is a strong indicator that these activities will perform bad in the validation data set, as by using a large number of features the classifier becomes easily overfit.

It is also shown that for the classification of a meeting only a single feature is required. This could indicate that meeting is easy to classify. However it seems more likely that is a result of the activity in the training set being too scripted or well performed. Due to this it has lost all relation with the natural activity of a meeting. As a result of this the meeting classifier will most likely perform worse in scenarios where natural meetings occur.

From Figure 9.8 similar properties can be derived. It is shown that, due to the large feature vector required, the interaction classification for faucet microwave and refrigerator will most likely have a low performance relative to the coffee pot.

9.3 Classifier optimization

To optimize the performance of the SVM the γ and C parameters need to be tuned. Intuitively, γ defines how far the influence of a single sample reaches, with low and high values meaning far and close respectively. The C parameter trades off misclassification of samples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly. A grid-search is most commonly used in the search for the optimal combination of γ and C. In this case a two stage search is performed.

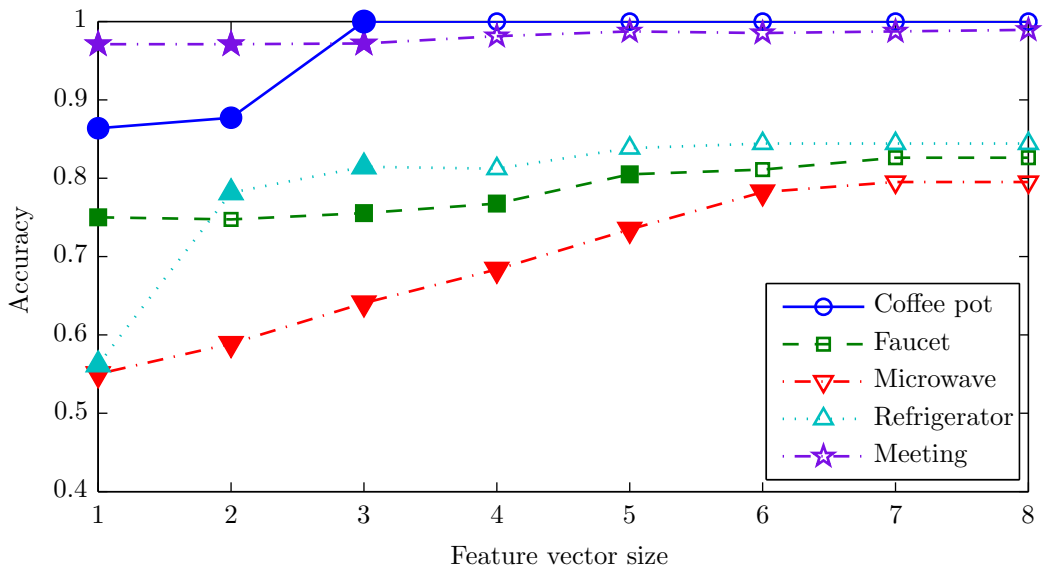


Figure 9.7: Accuracy vs. feature vector size for state classifier; based on 5-fold cross validation on training set, filled glyphs indicate selected features.

In the first stage coarse 2-D grid is traversed to find possible optimal combinations. During the second stage a fine grid is traversed around the combination with the highest performance in the coarse grid. These grid searches are usually performed on a logarithmic scale [34, 38], where log_2 and log_{10} are most commonly used. Here the log_2 is used with a coarse grid made up of $C = [2^{-5}, 2^{-3}, \ldots, 2^{13}]$ and $\gamma = [2^{-27}, 2^{-23}, \ldots, 2^{9}]$. During the search of the optimal parameter combination the standard definition from Matlab is used for the RBF kernel. This definition differs slightly from the general definition in that it uses σ instead of γ, where σ is defined as:

$$\sigma = \sqrt{\frac{1}{2\gamma}} \tag{9.8}$$

This means the meaning of σ is the inverse of the meaning of γ, so high values of σ indicate a far reaching influence and low values indicate a close reaching influence. Using Equation 9.8 the initial search range for σ is $\sigma = [2^{-5}, 2^{-3}, \ldots, 2^{13}]$. Using the optimal point, (σ_{opt}, C_{opt}), from the coarse grid a fine grid is constructed around it using $C = [2^{C_{opt}-2}, \ldots, 2^{C_{opt}}, 2^{C_{opt}+0.5}, \ldots, 2^{C_{opt}+2}]$ and $\sigma = [2^{\sigma_{opt}-2}, \ldots, 2^{\sigma_{opt}+2}]$. As a result the neighbourhood of the optimal point in the coarse grid is re-sampled with $4\times$ the resolution in the fine grid.

The performance of each point in the search grid is determined using 5-fold cross validation. This type of validation is needed to prevent the training from over-fitting the classifier. If the entire data set would be used the highest performance would by definition be at the highest value of C, as this would lead to the fewest misclassifications.

The average performance of all 5 iterations of the cross validation of each classifier is shown in Figure 9.9 to 9.13. All white areas in the figures have a performance less than the dark blue contour. Shown outlined is the inset of the fine grid in the coarse grid result. The

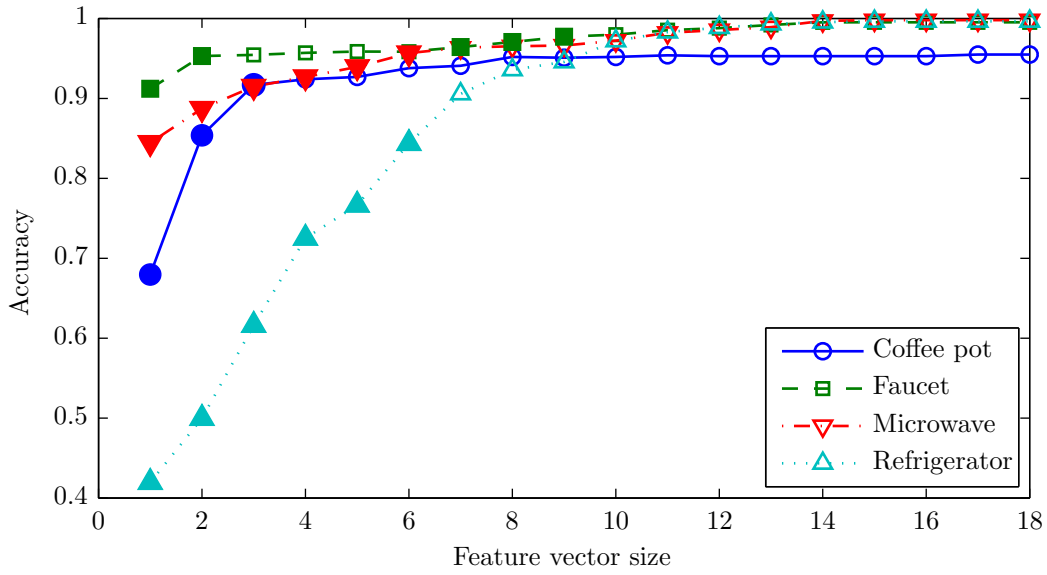


Figure 9.8: Accuracy vs. feature vector size for interaction classifier; based on 5-fold cross validation on training set, filled glyphs indicate selected features.

glyph indicates the location with the highest performance. Due to the difference in resolution between the two grids the contours do not lineup. In the figures, high values of C indicate a high degree of non-separable points between the classes. High values for σ indicate a large variance between the points in one class. Equation 9.7 is used to calculate the performance of a classifier.

The result of the grid search for the coffee pot classifiers is shown in Figure 9.9. The optimization of the state classifier shows that the classes are easily separable and the variance between the points in a class is low. For the interaction classifier, the classes contain some overlap resulting in a high C. However the performance difference between a high C and a low C is only 0.04, indicating only a very few samples are difficult to separate. There is no performance increase compared to the non-optimized state classifier, as both classifier have a performance of 1.0. The performance increase for the interaction classifier is 0.017.

The result of the grid search for the faucet classifiers is shown in Figure 9.10. Similar to the coffee pot the interaction classifier indicates some overlap between the classes resulting in a high C. However the performance difference between a high C and a low C is only 0.02, indicating only a very few samples are difficult to separate. The performance increase is 0.026 and 0.023 for the state and interaction classifier respectively.

The result of the grid search for the refrigerator classifiers is shown in Figure 9.11. The state classifier shows a clear separation between the classes, expressed in a very small C. This is unexpected as it was assumed that the separation between "off" and "cold" would be very bad. The interaction classifier has a very clear optimum at a medium C and a small σ. The large gradient in the x-direction indicates that not only is the variance inside each class small, the distance between the two classes is also small compared to the class variance. This could lead to problems in the validation data set if as a small shift in the mean of the class would result in a large amount of misclassified samples. The performance increase is 0.063 and 0.051

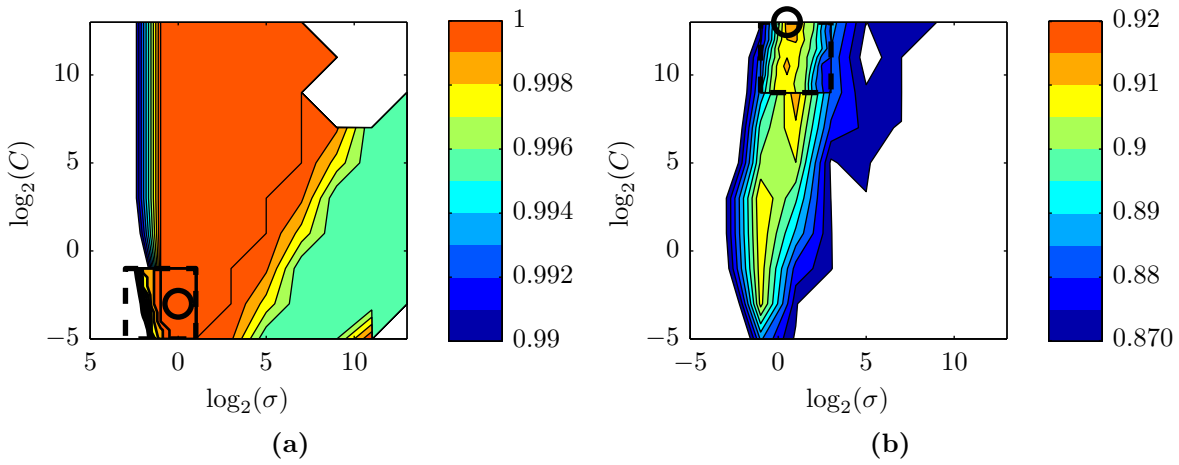


Figure 9.9: Coffee pot classifier tuning using grid-search and 5-fold cross validation on training set; (a) state classifier - maximum performance: 1.0, (b) interaction classifier - maximum performance: 0.913.

for the state and interaction classifier respectively.

The result of the grid search for the refrigerator classifiers is shown in Figure 9.12. Both the state and interaction classifier show a large degree of overlap between the classes. As perviously states this is due to the very small perceivable differences between the different states of the microwave. The performance increase is 0.092 and 0.035 for the state and interaction classifier respectively.

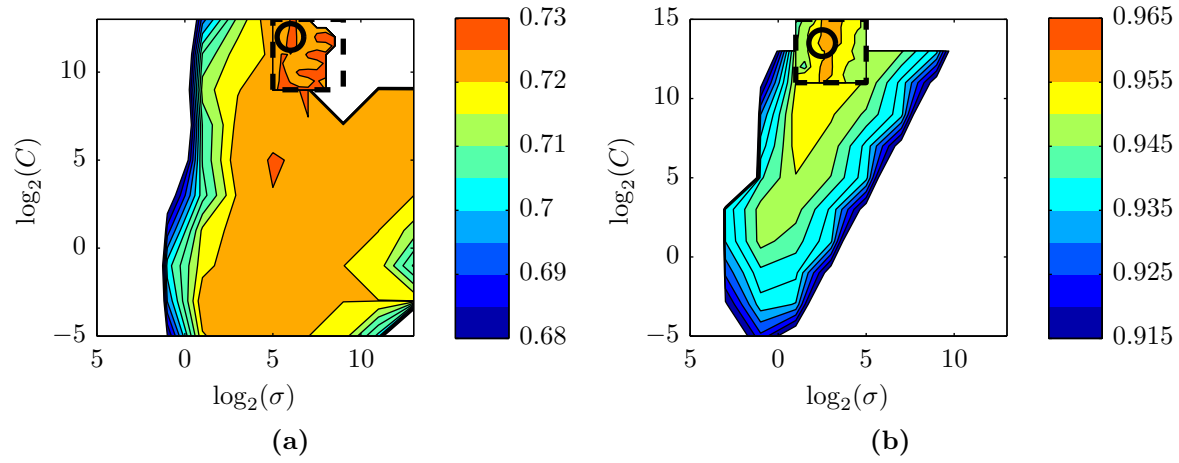


Figure 9.10: Faucet classifier tuning using grid-search and 5-fold cross validation on training set; (a) state classifier - maximum performance: 0.732, (b) interaction classifier - maximum performance: 0.962.

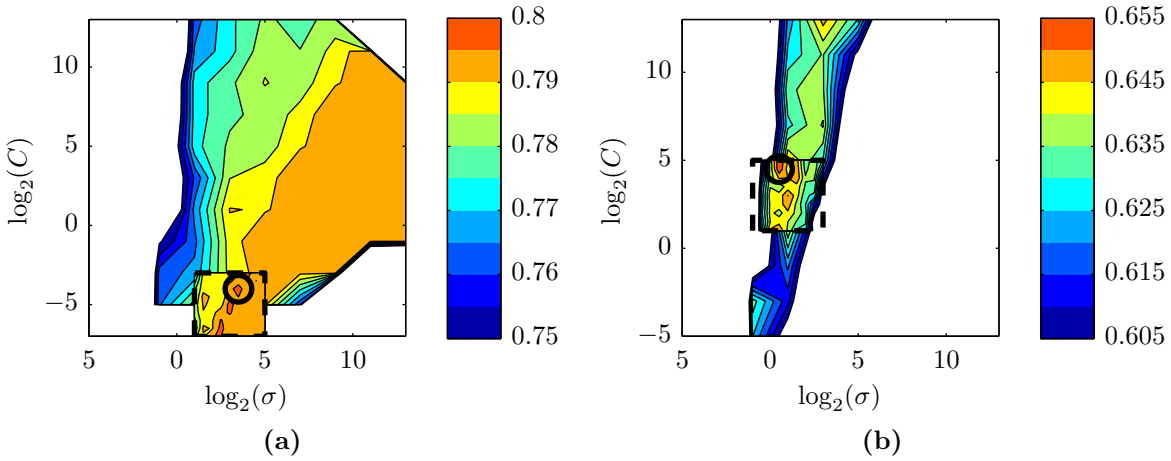


Figure 9.11: Refrigerator classifier tuning using grid-search and 5-fold cross validation on training set; (a) state classifier - maximum performance: 0.797, (b) interaction classifier - maximum performance: 0.654.

The result of the grid search for the meeting classifiers is shown in Figure 9.13. The performance difference between a small and large C is only small, indicating only a few samples overlap between the classes. The performance increase is 0.003.

The obtained optimal tuning parameters are used to train the final classifiers.

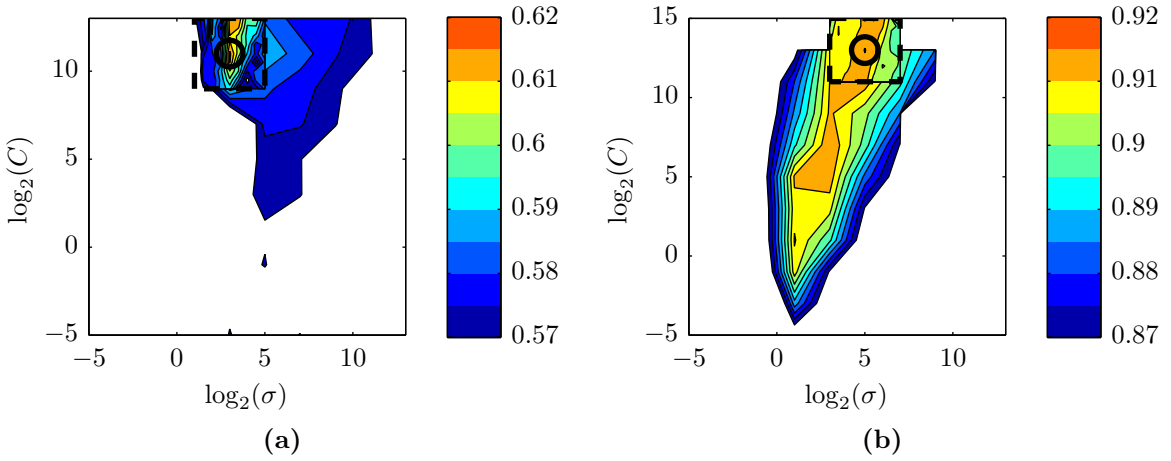


Figure 9.12: Microwave classifier tuning using grid-search and 5-fold cross validation on training set; (a) state classifier - maximum performance: 0.617, (b) interaction classifier - maximum performance: 0.915.

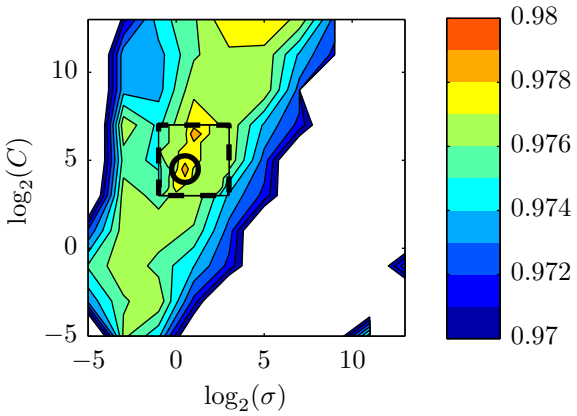


Figure 9.13: Meeting state classifier tuning using grid-search and 5-fold cross validation on training set, maximum performance: 0.979.

Table 9.2: Accuracy classification training data set.

	State	Interaction
Coffee pot	1.00	0.88
Faucet	0.82	0.96
Microwave	0.67	0.92
Refrigerator	0.79	0.88
Meeting	—	0.97

9.4 Training result

Using the optimal tuning parameters obtained in Section 9.3 the classifiers are trained on the entire training data set. The resulting SVMs will then be used in the interaction classification module, before the output filter. To quantify the performance of the classifiers, accuracy is used as defined in Equation 9.7 For multi-class classifiers the accuracy is averaged over different binary classifiers.

The accuracy for the classifiers is listed in Table 9.2. The second column lists the accuracy of the state classifiers and the third column the accuracy of the interaction classifiers. As expected the accuracy of the microwave state classifier is very low. The obtained results agree with the accuracy which was obtained during tuning.

Where the results presented in this chapter are obtained by classifying the individual samples from the isolated activity patterns, the results presented in Chapter 10 are obtained by running the framework on the continuous data sets.

9.5 Framework performance

Using the optimally trained classifiers from Section 9.4, the performance of framework is determined for the 21 training sets. The performance is first determined with the output filter disabled. This test gives a baseline performance. Using this baseline a sweep is made

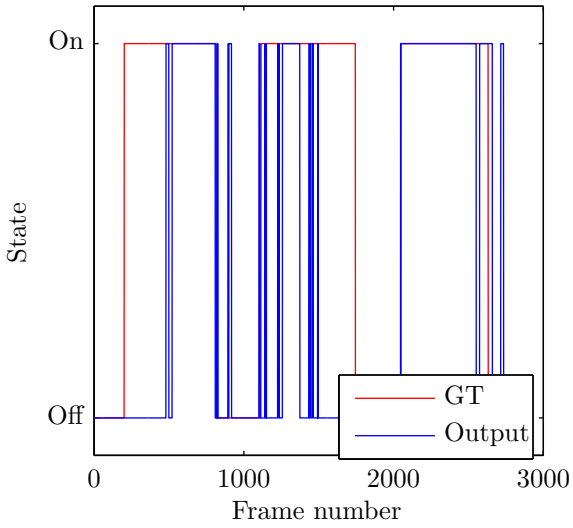


Figure 9.14: Output of the framework, without output filter, for the microwave state, using training data set 8, 9 and 10.

over the window size of the output filter. Using the result of the sweep the optimal window size is determined.

The initial baseline performance is listed in Table 9.3. The framework without an output filter has a very poor performance. While the recall values are very high, 1.00 for all but 3 actions, the precision is very low. A precision of 0.14 for the microwave interaction effectively means it is impossible to distinguish if somebody is using the microwave or only standing next to it. The low precision of all classifiers is caused by the large amount of chatter which is present in the output of the framework.

The calculated output for microwave state based on training sets 8, 9 and 10 is shown in Figure 9.14. Especially in the first two training sets the chatter is clearly visible. For a period of 5 to 10 seconds around the state change the output of the framework jumps between "on" and "off" before it settles at the correct state. Also visible is the difference in the way the first two data sets are recorded compared to the last data set. For the first two data set the person remained stationary in front of the microwave, whereas for the last data set the person moved a meter back from the microwave. This moving back eliminates the interference between the heat signature of the person and the microwave. This interference of a person standing close to the microwave increases the temperature variance of the microwave, which results in an "off" state classification. This is because a high temperature variance is associated in the training set with interaction, meaning either placing objects in or taking objects out of the microwave. As the microwave is always off during interactions the classifier is trained to classify high temperature variance as "off". Similar reasoning holds for the faucet and the refrigerator. For the faucet the thermal difference between off and cold water is very small, resulting in a large amount of chatter between the two states. For the refrigerator the difference between open and interacting (opening or closing) is very small. As a result the classifier continuously switches between the "open" and "closed" state.

To remove the chatter around the state changes the output filter described in Section 7.2 is enabled. To determine the optimal window size for the output filter a sweep is performed.

Table 9.3: Normalized accuracy classification training data set using no output filter.

	State		Interaction	
Coffee pot	1.00	1.00	0.60	1.00
Faucet	0.18	0.93	0.17	1.00
Microwave	0.17	1.00	0.14	1.00
Refrigerator	0.35	1.00	0.25	0.79
Meeting	—	—	0.33	0.77

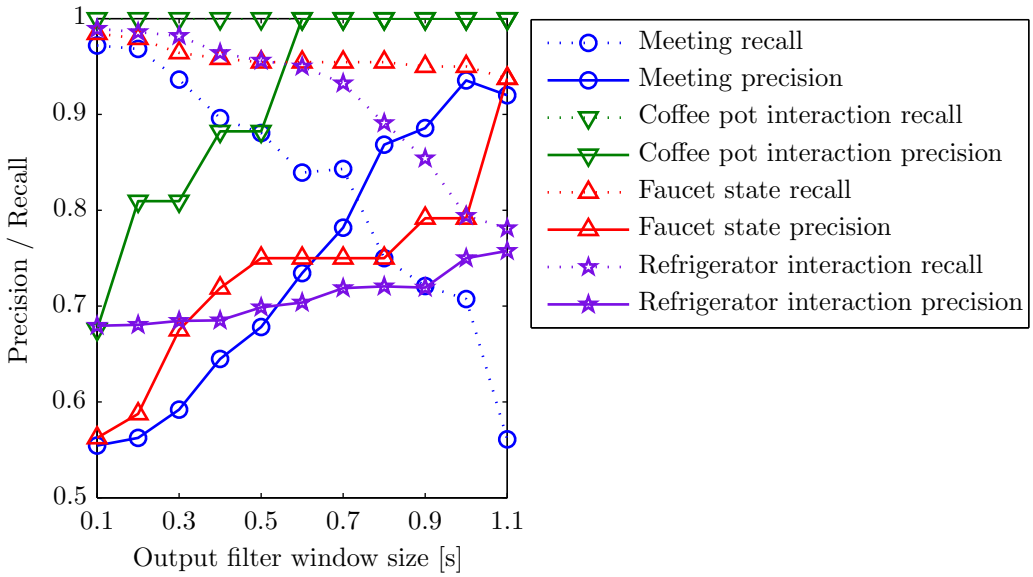


Figure 9.15: Precision recall plot training data set; output filter window size swept between 0.1 s and 1.1 s.

The sweep is performed for a filter length of 0.1 s to 1.1 s. Extending the filter further would create an unacceptably large delay compared to the duration of some activities.

The results of the sweep are shown in Figure 9.15. From the figure the classifiers can be, based on their gradients, divided into four distinct groups. These four groups consist of the following classifiers.

- Refrigerator interaction, which has a small gradient. By increasing the filter window size instead of noise, correctly classified activities are removed from the output. This indicates that the sequence length of correctly classified activities is smaller than that of the noise. As a result, increasing the window size to very large lengths, e.g. 300 samples, will only result in loss of recall and only marginal increase in precision.
- Meeting interaction, which has a medium gradient. Here the filter removes noise and correctly classified activities equally.
- Faucet state, which has a large gradient. Here increasing the filter window size removes

Table 9.4: Classification results training data set using optimal output filter.

		Filter window	P	R
Coffee pot	State	< 0.1	1.00	1.00
	Interaction	0.6	1.00	1.00
Faucet	State	1.1	0.94	0.94
Microwave	State	$\gg 1.1$	0.58	1.00
Refrigerator	State	$\gg 1.1$	0.55	1.00
	Interaction	1.1	0.75	0.78
Meeting	Interaction	0.7	0.86	0.75

primarily only noise, only a small fraction of correctly classified activities are removed.

- Coffee pot interaction, microwave state, refrigerator state and interaction, which all have an infinitely large gradient. Here increasing the filter window size is very effective as it only removes noise. As a result the precision is increased without any loss in recall.

The optimal window size differs per classifier. The optimal window size for each classifier is listed in Table 9.4. The coffee pot state requires no filter at all. However this is due to the training set. In all training data sets the coffee pot has either been on for a long period or off for long period. There is no training set where the coffee pot has a transition between on and off.

The coffee pot interaction requires a small filter. This is due to chatter between the "present" and "away" state. The faucet requires a long filter window of 1.1 seconds. Using such a long filter results in a high precision of 0.94 as it eliminates the chatter between "cold" and "off". The performance of the refrigerator and the microwave is, even with the output filter, low. As stated above, a very large filter window would be required in order to obtain an optimal performance for these classifiers. However, a very large filter window is unpractical due to its large delay. Furthermore, due to the implementation of the filter, all classifiers use the same filter length. All objects with a lifetime shorter than the filter length are not considered for the output of the framework. This means that, if the filter window is set too large the performance for short activities is degraded. For example, if the windows size is set to 30 seconds, anytime somebody would get a cup of coffee and place the pot back under 30 seconds would not show up in the output of the framework.

The meeting classifier has its optimum at a window size of 0.7 seconds. Most of the chatter for the meeting classifier has a very high frequency. It is almost exclusively caused by a single person which, due to noise, increases its size for only single or two frames. The remainder of the chatter is caused during interaction of a single person with the coffee pot. The increase in size due to moving the hot pot is misclassified as a meeting between to people.

The resulting optimal filter window size which will be used for all classifiers is determined to be 0.7 seconds. Using the optimal window size the performance of the framework for the validation data set is determined.

9.6 ROC curves

For Table 9.2 accuracy is used as a performance metric. A downside of accuracy as a metric is that it is suspectable to class skew [39]. A result of this is that the performance of the framework can be improved by only adding empty frames (frames with no objects present) to the data set. If enough empty frames are added it is possible to obtain an accuracy of 100%. To remove this dependence on class skew, receiver operating characteristic (ROC) curves are created for the classifiers. ROC curves depict the tradeoff between true positive rate (TPR) and false positive rate (FPR). TPR and FPR are defined using the following equations:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9.9}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + TN} \tag{9.10}$$

where TP and FP are the number of true positives and false positives respectively and TN and FN are the number of true negatives and false negatives respectively.

The implementation used to create the ROC plots is based on [40]. In order to create a ROC curve a threshold is varied between two extremes. One extreme is where the classifier only produces class -1 as its output, the other extreme is where the classifier only produces class 1 as its output. As a SVM is a non-probabilistic classifier it does not contain a decision threshold and thus a ROC curve can not directly be created. It is important to note that the decision hyper-plane used by the SVM is not the same as a threshold. This is because the hyper-plane is only the optimal separation between the two classes, it contains no probabilistic information. Using the bias of the SVM to create a ROC curve is wrong, as the bias only contains information about the class skew and to some extent the size of the training data set.

In order to make the output of the SVMs suitable to create a ROC curve binary logistic regression is performed on the output of the classifier. This is similar to what is done in [40]. Using the logistic regression a probability is assigned to the output of the SVM. The probability is based on the distance between the sample point and the classification hyper-plane. Conceptually, points which are located further from the hyper-plane have a higher probability of being correctly classified than points which are located closer. To train the binary logistic regression model six steps are performed.

1. The training data set is split into two sets.
2. The SVM is trained on one set.
3. Using the SVM the distance to the hyper-plane is determined for all samples in the other set.
4. Divide the data points used in step 3 into bins based on there distance to the hyper-plane.
5. Determine for each bin the ratio of correctly classified data points.
6. Train the regression model on the probabilities obtained in step 5.

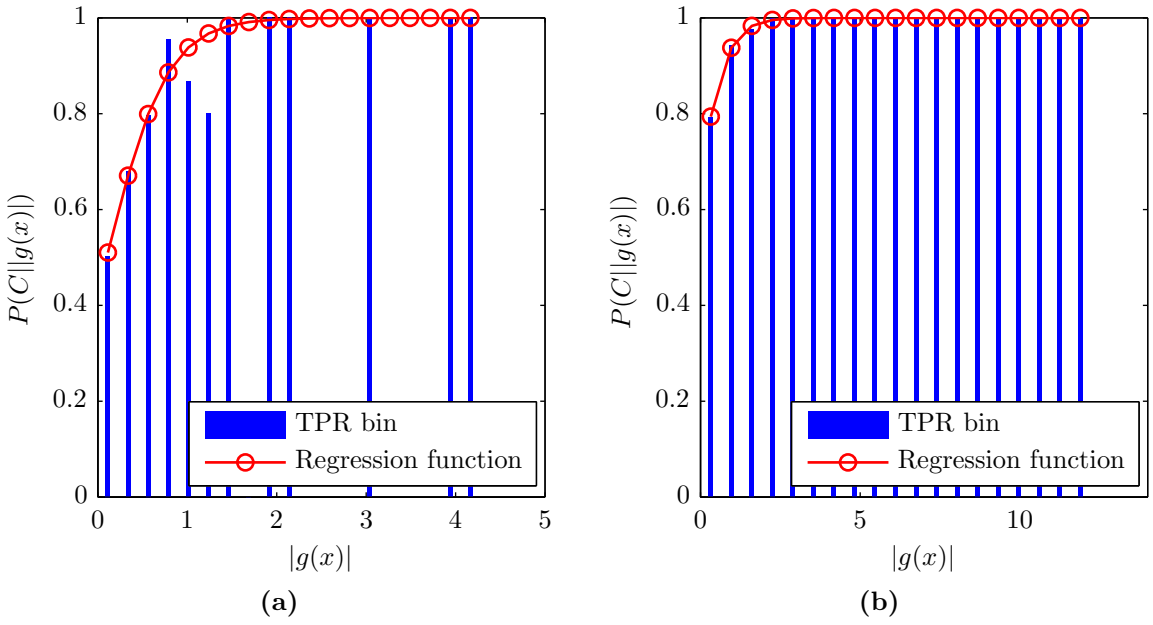


Figure 9.16: Binary logistic regression function microwave; (a) state classifier, (b) interaction classifier.

The number of bins used in step 4 is an important parameter for training the regression model. If too few bins are used the model will be insensitive, if too many bins are used the model is susceptible to noise. In the framework 20 bins are used to train to model. This has proven to provide good results. Bins which contain no sample points are not used in step 6. Logit is used as link function for the regression model. Figure 9.16a and 9.16b show the output of step 5 and 6 for the microwave state and interaction classifier respectively. In Figure 9.16a it is shown that points which are located close to the hyper-plane have a low probability of being correct, $P(C||g(x)|) \approx 0.5$, while points which are located far away have a probability of 1 for being classified correctly. This matches the assumption made earlier.

The ROC curve for the refrigerator interaction classifier is shown in Figure 9.17. From the blue plot in the figure it can be concluded that the "present vs. interacting" classifier will have a low performance. It performs only a little better than flipping a coin, which is indicated with the red plot. The output of the classifier, shown in green, has, despite this, a reasonable performance. This is because the present and interacting activity are merged to a single activity for the output, as was specified in 8.8. The red circle in Figure 9.17 approximately indicates the performance of the SVM without regression. The ROC plots for the refrigerator and microwave state are shown in Figure 9.18 and Figure 9.19 respectively. Next to the ROC curves for the SVM with regression output, the figures also show the ROC curves for the state machine output and the ALF. The state machine used to create these figures is a simplification of the HMMs described in Chapter 7.2. The state machine used limits the states using a priori information. No training is used in the generation of the state machines. In essence the state machine retains its state when nobody is present near the appliance. Meaning the appliance can only change state if a person is present.

From both figures the same conclusion can be drawn. By adding the state machine the

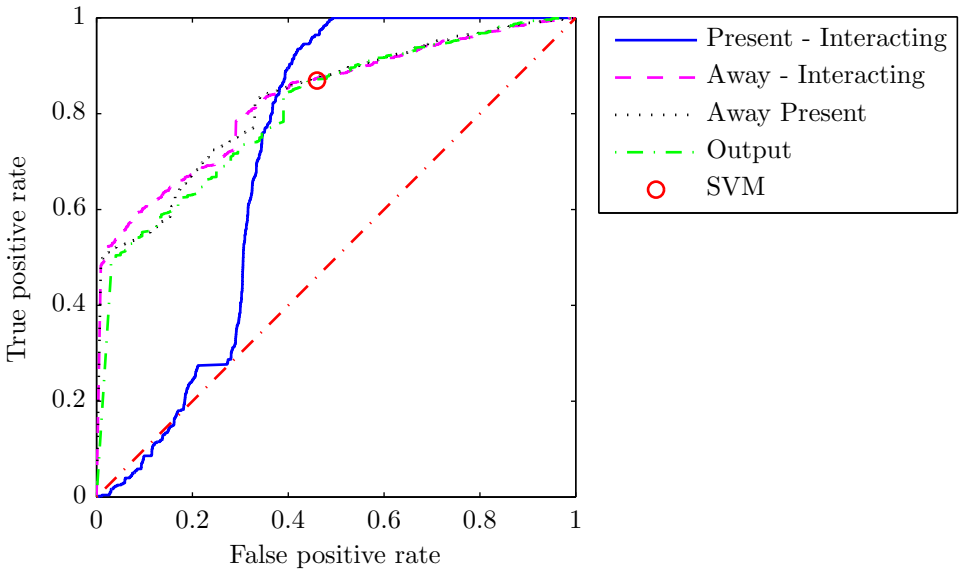


Figure 9.17: ROC curve refrigerator interaction.

performance of the classifiers is reduced. This reduction of performance is probably caused by the fact that, once nobody is present near the appliance, an erroneous state is propagated through the framework until somebody again comes near the appliance. This has as result that the system cannot recover from a misclassification performed when a person is leaving the scene, as it is inhibited by the state machine. As a different state machine is used compared to the HMMs the results presented in the ROC curves are not directly comparable to the accuracies presented elsewhere in the report.

A solution to the propagation problem would be to incorporate some form of recovery mechanism in the state machine. This could be in the form of an accumulator which uses the classification of multiple frames to better classify the state of the appliance.

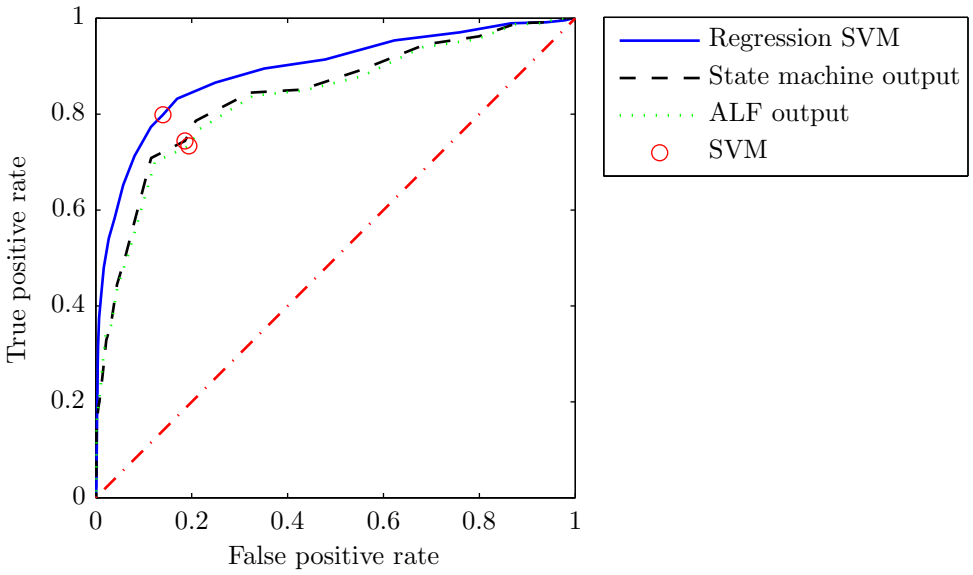


Figure 9.18: ROC curve refrigerator state.

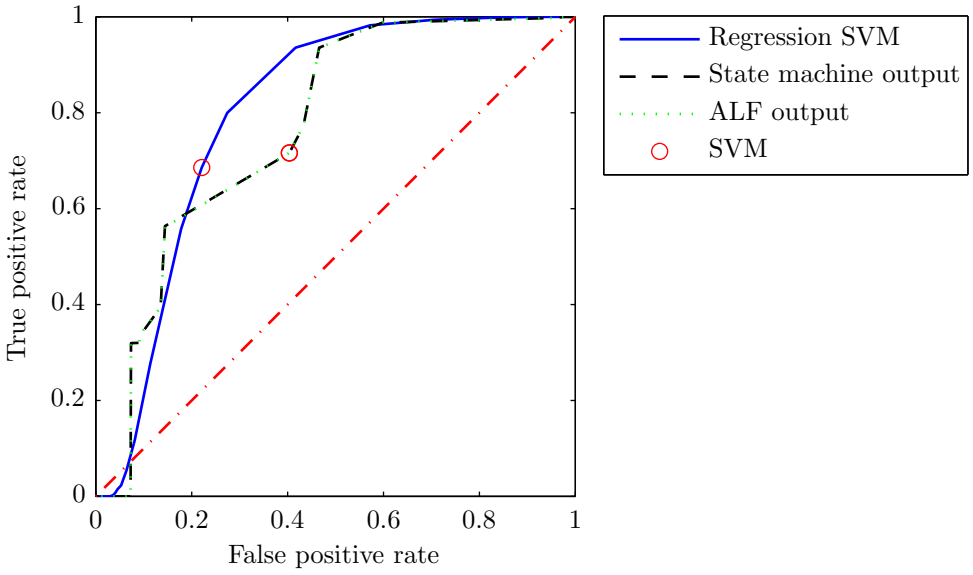


Figure 9.19: ROC curve microwave state.

Chapter 10

Results

This chapter presents the results obtained from running the framework on the continuous validation data set, without the use of additional segmentation markers. Using the validation data set the performance of the framework, using tuned classifiers and optimized output filter length, is determined. The performance of the framework in combination with the validation data set is a good indicator of the robustness of the framework.

Table 10.1 lists the precision, recall and accuracy values for the different activities. The precision of the coffee pot interaction is very good, even with the pot slightly moved compared to the training set. Also all activities performed with the electric kettle, instead of the coffee pot, are correctly classified. This shows that the framework is very robust against small displacements of appliances. It also shows that the particular appliance used for the activity is not important, the only requirement is that they have a similar thermal signatures.

The performance of the faucet is lower than would be expected from the training data set. This is caused by the difference in the way the activities are performed in the validation data set compared to the training data set. In the validation data set people fill a plastic bottle or only quickly rinse their coffee cup with cold water. Both of these activities have a very different thermal signature compared with the training set, where water was extensively splashed around in the sink. Especially, the filling of containers with water is difficult to detect. The containers only have a small projection on the sensor. The small temperature difference induced by the container is completely negated by the hand holding it. Furthermore, the GT notation for the faucet is also has a low accuracy due to the mounting location of the control camera, possibly resulting in an even worse accuracy for the classifier.

The performance of the microwave is lower than expected. This is caused by the presence of a bulletin-board above the microwave. People reading the bulletin-board are incorrectly classified as using the microwave. The performance of refrigerator state is as expected, however the performance of the refrigerator interaction is much lower. This is caused by the inability of the classifier to distinguish between somebody moving in front of the refrigerator and actually opening it.

The performance of the meeting classifier is also lower than expected. This low performance has two reasons. People who walk around with the coffee carafe, to for example serve the coffee near the refrigerator instead of the coffee pot, are incorrectly classified. Also the GT notation for a meeting is extremely difficult to get correct. This is the result of the definition of a meeting, which is almost impossible to annotate. The definition requires the annotation to indicate, based on a picture showing the side view of the scene, if there are

Table 10.1: Classification results validation data set.

(a)

		P	R
Coffee pot	State	1.00	1.00
	Interaction	0.97	1.00
Faucet	State	0.63	0.72
Microwave	State	0.33	1.00
Refrigerator	State	0.57	0.79
	Interaction	0.18	0.64
Meeting	Interaction	0.64	0.77

(b)

	Activity		Accuracy
Coffee pot	State	Off	—
		On	1.00
	Interaction	Away / Present	0.97
		Serving	0.96
Faucet	State	Off	0.80
		Hot / Cold	0.45
Microwave	State	Off	0.34
		On	0.32
Refrigerator	State	Closed	0.67
		Open	0.50
	Interaction	Away /Present	0.24
		Interacting	0.11
Meeting	Interaction	No	0.66
		Yes	0.61

multiple dynamic objects in the field of view which will form a single large object in the thermal image.

The results for the validation data set show that the developed framework is robust against small variances in the performed actions. It also shows that the sensor is well suited for the classification of activities which have a clear thermal signature, like the coffee pot. For appliance which lack a clearly visible thermal signature, like the cold water from the faucet, an open refrigerator and microwave which is turned on, additional features, or possibly sensor modalities, are required to obtain an acceptable performance level. The distance and position variance features of the dynamic objects are not enough. From the result in Table 10.1 it seems the sensor is not well suited for detecting meetings. On the validation data set the sensor obtained a 98.2% accuracy for detecting dynamic objects. The low accuracy for detecting meetings is therefore, more the result of a too difficult to annotate definition then a shortcoming of the sensor itself.

Chapter 11

Discussion

In this chapter first the overall conclusion for this thesis is presented. Next some recommendation are given to provide starting points for possible extension or improvement of the developed framework.

11.1 Conclusion

It is shown that the used 8×8 thermopile provides enough information to detect complex activates, like serving a cup of coffee or using the faucet. With the test data set it is shown that the grid configuration allows for easier mounting compared to traditional motion sensors, as a relatively complex areas can be covered with one sensor and without the need of carefully measuring the overlap of each thermal device.

It is shown that a single sensor installation can provide information on various activities, which would normally require instrumenting many devices and appliances with individual sensors.

The following contributions are made in this thesis:

1. The thermopile sensing concept is introduced and a novel processing framework is developed to process the sensor data. The framework detects and tracks objects in the sensors field of view. Additional a priori information is used to aid in the detection of objects which due to their state cannot be independently detected by the sensor. The state of and the interactions between the different detected objects are classified into different categories. The developed framework provides concurrent responses for all detected objects, thus can process multi-user scenarios.

2. An evaluation is performed using (1) a scripted set of 21 activities used as a training data set, and (2) an unscripted, real-life study data set used for testing. During the training analysis, optimal features and parameter sets for the classifiers were determined.

3. The framework is evaluated using the real-life unscripted data set and the classification performances for all concurrent state and interaction classifiers is determined.

From the evaluation of the framework on the validation data set it is shown that the used classification scheme is insensitive to small variations in the performed activities. It is also shown that the framework obtains an excellent performance of 0.95 for activities with a clear thermal signature, like the coffee pot. For appliances of which the state can only be

inferred from circumstantial evidence the performance is relatively low, ranging from 11.4% for detecting opening and closing of the refrigerator to 0.46 for detecting whether the faucet is off or cold water is being used. The object detection algorithm show a very good accuracy of 0.98. This indicates that the sensor is very well suited to be used to detect presence and activities which have a clear thermal signature.

It can be expected that every recognized activity can be mapped to an energy cost, which in turn, can be fed back to the user to guide energy consumption awareness. Due to the design of the framework energy consumption feedback can be given instantaneously, less 1 second delay, when the activity is being performed.

In short, the developed framework allows for accurate classification of interactions and state of appliances using an 8×8 thermopile matrix sensor. The work presented in this thesis has resulted in a paper, which is accepted for publication in the SEIT 2013[1] conference.

11.2 Future work

The developed framework heavily relies on the accurate detection of the objects inside field of view of the sensor. Although the current proposed and implemented algorithms result in an overall good performance of the framework, same measures can be taken to improve the performance of the specific classifiers.

To improve the performance of the meeting classifier the solidity or robustness of created objects needs to be further increased. This increase in robustness would prevent the creation of super-objects joined only by a single, possible, noisy pixel. A standard technique to increase robustness is to apply morphological erosion and dilation on the output of the connected component labeling algorithm. If a suitable structuring element is used for the erosion the single noisy pixel could be removed. Through dilation the two individual objects are then reconstructed. Most likely, apart from the occupancy map, thermal information must be included in this process to prevent valid objects from being removed by the erosion step.

The refrigerator and microwave classifier could be improved by repositioning the sensor. If the sensor is positioned so it can see inside the refrigerator when the door is opened, it would greatly increase not only the performance of the state classifier but also the interaction classifier. A possible suitable location would be between marker F and H in Figure 8.2. In this location the sensor still has a clear view of the faucet and coffee pot. However it gains the advantage that it can see the inside of the door of the refrigerator.

To reduce the complexity of the classification module, the output filter could be replaced by a simple state limiting low pass filter. Furthermore the filter length of the activity length filter could be optimized for each activity individually.

To highlight the temporal constraints in the activities, additional features could be selected which better represent these constrains. An example would be the temperature variation of an object in time.

It could be interesting to use the Grid-EYE sensor in combination with an external lens. Using the lens it would be possible to change the grid to have it better suit the optimal layout for the classification task at hand. An example would be classify different modes of desk work by changing the layout to a rectangular grid to match that of the desk.

[1] International Conference on Sustainable Energy Information Technology (SEIT) 2013 - http://cs-conferences.acadiau.ca/SEIT-13/

The current implementation of the framework runs off-line. A next step would be to change the architecture to an online environment, with the possibility of running it on the same processing unit (Arduino platform) which is also used to interface with the sensor. This means the framework should be fine tuned for low resources and power consumption.

List of Figures

List of Tables

Bibliography

[1] N. Oliver, A. Garg, and E. Horvitz, "Layered representations for learning and inferring office activity from multiple sensory channels," *Comput. Vis. Image Underst.*, vol. 96, no. 2, pp. 163–180, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2004.02.004

[2] D. Kawanaka, T. Okatani, and K. Deguchi, "Hhmm based recognition of human activity*this paper was presented at mva2005." *IEICE - Trans. Inf. Syst.*, vol. E89-D, no. 7, pp. 2180–2185, Jul. 2006. [Online]. Available: http://dx.doi.org/10.1093/ietisy/e89-d.7.2180

[3] C. R. Wren and E. M. Tapia, "Toward scalable activity recognition for sensor networks," in *In Lecture Notes in Computer Science*. Springer, 2006, pp. 168–185.

[4] F. Wahl, M. Milenkovic, and O. Amft, "A distributed pir-based approach for estimating people count in office environments," in *Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on*, dec. 2012, pp. 640 –647.

[5] k. Murao, T. Terada, A. Yano, and R. Matsukura, "Detecting room-to-room movement by passive infrared sensors in home environments," in *AwareCast 2012: Workshop on Recent Advances in Behavior Prediction and Pro-active Pervasive Computing*, 2012.

[6] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *In Pervasive*, 2004, pp. 158–175.

[7] D. Linzmeier, "Pedestrian detection with thermopiles using an occupancy grid," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, oct. 2004, pp. 1063 – 1068.

[8] T. Gindele, S. Brechtel, J. Schroder, and R. Dillmann, "Bayesian occupancy grid filter for dynamic environments using prior map knowledge," in *Intelligent Vehicles Symposium, 2009 IEEE*, june 2009, pp. 669 –676.

[9] J. Davis and V. Sharma, "Robust background-subtraction for person detection in thermal imagery," in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, june 2004, p. 128.

[10] D. Ramanan, D. Forsyth, and A. Zisserman, "Strike a pose: tracking people by finding stylized poses," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, june 2005, pp. 271 – 278 vol. 1.

[11] L. Trujillo, G. Olague, R. Hammoud, and B. Hernandez, "Automatic feature localization in thermal images for facial expression recognition," in *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, june 2005, p. 14.

[12] W. Wang, J. Zhang, and C. Shen, "Improved human detection and classification in thermal images," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, sept. 2010, pp. 2313 –2316.

[13] J. Honorato, I. Spiniak, and M. Torres-Torriti, "Human detection using thermopiles," in *Robotic Symposium, 2008. LARS '08. IEEE Latin American*, oct. 2008, pp. 151 –157.

[14] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 150–161. [Online]. Available: http://doi.acm.org/10.1145/958491.958509

[15] S. Oh and S. Sastry, "Tracking on a graph," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1147685.1147721

[16] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham, "On target tracking with binary proximity sensors," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, April, pp. 301–308.

[17] X. Liu, G. Zhao, and X. Ma, "Target localization and tracking in noisy binary sensor networks with known spatial topology," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2, April, pp. II–1029–II–1032.

[18] T. Teixeira, G. Dublon, and A. Savvides, "A survey of human-sensing: Methods for detecting presence, count, location, track, and identity," 2010.

[19] N. Shrivastava, "Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms," in *in SenSys 06: Proc. 4th Internat. Conf. on Embedded Networked Sensor Systems, 2006*. ACM Press, 2006, pp. 251–264.

[20] X. Luo, B. Shen, X. Guo, G. Luo, and G. Wang, "Human tracking using ceiling pyroelectric infrared sensors," in *Control and Automation, 2009. ICCA 2009. IEEE International Conference on*, Dec., pp. 1716–1721.

[21] Texas Instruments, "Ultra-low power motion detection using the msp430f2013," March 2009. [Online]. Available: http://www.ti.com/lit/an/slaa283a/slaa283a.pdf

[22] J. Fraden, *Handbook of modern sensors physics, designs, and applications*. New York: AIP Press/Springer, 2004.

[23] C. Kittel, *Thermal physics*. San Francisco: W. H. Freeman, 1980.

[24] Panasonic Electric Works Corporation, "Infrared array sensor: Grid-eye." June 2012. [Online]. Available: http://pewa.panasonic.com/assets/pcsd/catalog/grid-eye-catalog.pdf

[25] I. T. Center, "R&d basic coarse; publication t560455_a-en-us," 2004.

[26] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE PAMI, 1991*, vol. 13, no. 6, pp. 583–598, 1991.

[27] D. Brown, "Decentring distortion of lenses." *Photogrammetric Engineering*, vol. 32, 1966.

[28] D. Linzmeier, D. Vogt, and P. Prasanna, "Probabilistic signal interpretation methods for a thermopile pedestrian detection system," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, june 2005, pp. 12 – 17.

[29] R. S. Merali and T. D. Barfoot, "Patch map: A benchmark for occupancy grid algorithm evaluation," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, oct. 2012, pp. 3481 –3488.

[30] R. M. Gray and L. D. Davisson, *An introduction to statistical signal processing*. Cambridge University Press, 2004.

[31] E. Lee, "Region filling using two dimensional grammars," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, mar 1987, pp. 1475 – 1478.

[32] S. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory (v. 1)*. Prentice Hall, 1993.

[33] T. Yang, Q. Pan, J. Li, and S. Li, "Real-time multiple objects tracking with occlusion handling in dynamic scenes," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, june 2005, pp. 970 – 975 vol. 1.

[34] L. Hermes and J. Buhmann, "Feature selection for support vector machines," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 2, 2000, pp. 712 –715 vol.2.

[35] A. Maurya, "Support vector machines: What is the intuition behind gaussian kernel in svm." Februari 2013. [Online]. Available: http://www.quora.com/Support-Vector-Machines/What-is-the-intuition-behind-Gaussian-kernel-in-SVM

[36] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 –286, feb 1989.

[37] R. Durbin, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[38] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, "A practical guide to support vector classification," 2003.

[39] M. Stager, P. Lukowicz, and G. Troster, "Dealing with class skew in context recognition," *2012 32nd International Conference on Distributed Computing Systems Workshops*, vol. 0, p. 58, 2006.

[40] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[41] muRata, "Pyroelectric infrared sensors," October 2012. [Online]. Available: http://www.murata.com/products/catalog/pdf/s21e.pdf

[42] ——, "Smallest and thinnest in the world! developing surface mount pyroelectric infrared sensor," February 2010. [Online]. Available: http://www.murata.com/new/news_release/2010/0226/index.html

[43] Hamamatsu, "Technical information sd-12, characteristics and use of infrared detectors," November2 2004. [Online]. Available: http://sales.hamamatsu.com/assets/applications/SSD/Characteristics_and_use_of_infrared_detectors.pdf

[44] Thermoworks, "Thermoworks emissivity table." [Online]. Available: http://www.thermoworks.com/emissivity_table.html

[45] Agilent Technologies, "Agilent 34970a data acquisition/switch unit family," October 2012. [Online]. Available: http://cp.literature.agilent.com/litweb/pdf/5965-5290EN.pdf

Appendix A

Pyroelectric sensor qualification

This chapter investigates the suitability of a pyroelectric sensor as a true presence detector. As sensor the IRS-B210 pyroelectric infrared dual element sensor from muRata is used, see Figure A.1. The main features of the sensor are [41]:

- Reflow surface-mounting support
- Smallest and ultra-thin through the trade ($4.7 \times 4.7 \times 2.4\,\text{mm}$)
- High sensitivity
- Achieves superior electromagnetic noise resistance characteristics

This sensor is chosen as it is currently being used in a product which is designed and mass-produced by AME. This chapter has the following outline. Section A.1 describes the theoretical model of a pyroelectric crystal and more generally a pyroelectric sensor, section A.2 describes the experiments performed with the sensor to determine its suitability for presence detection. In section A.3 a conclusion in presented regarding the sensors' suitability.

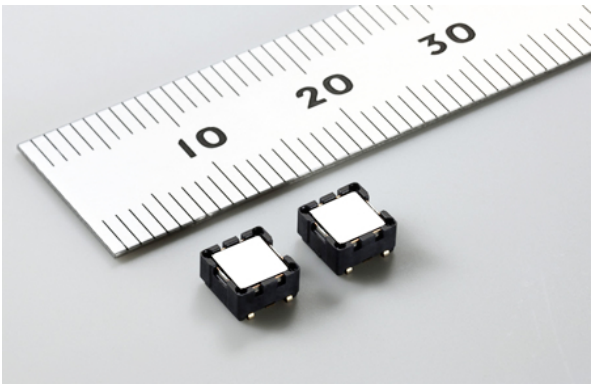


Figure A.1: IRS-B210 from muRata [42].

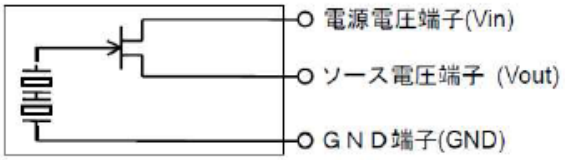


Figure A.2: Equivalent circuit [42].

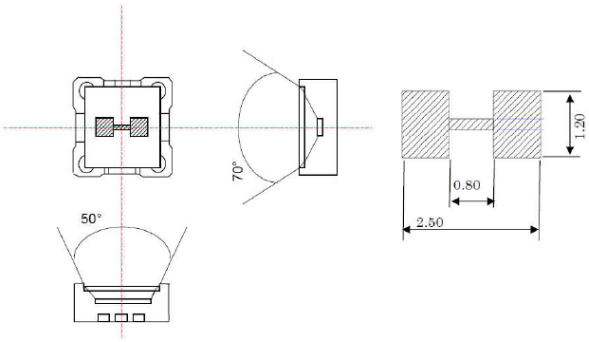


Figure A.3: Internal element geometry and field of view [42].

A.1 Model

The used sensor is of the type *dual element*. This means the sensor contains two pyroelectric active areas connected in buck mode. This has the advantage that the two elements cancel each other out in the case they receive the same amount of IR radiation, or in other words the output equals the difference between the two elements. This makes sensor only sensitive for objects of which the coverage is different for the two active areas.

As all pyroelectric elements are also piezoelectric, the sensor elements are susceptible to motion. Another advantage of a dual element sensor is that this susceptibility to motion is greatly reduced, because if the motion vector is equal for both elements they will counteract each other due to the buck configuration.

Figure A.2 shows the equivalent circuit for a generic dual element sensor. This circuit is similar for all dual element sensors, independent of manufacturer, which use a JFET as transimpedance converter and have a voltage output. As opposed to sensor which use an operational amplifier as transimpedance converter and have a current output. It is therefore also applicable for the sensor used during these experiments. The main difference between sensors with current and voltage output is their sensitivity with respect to the velocity of the object. For sensor with current output their normalized sensitivity is usually low, however it is constant with respect to the velocity. On the other hand the normalized sensitivity of sensors with a voltage output is high but is frequency depended.

The internal structure of the sensor is shown in Figure A.3. The sensor has a field of view of 70 ° in one axis and 50 ° in the other. The elements each measure $1.2 \times 0.85\,\text{mm}$. During the experiments no additional lenses are used. Figure A.4 shows the sensitivity of an element with respect to velocity or chopping frequency. R_g represents the combined resistance measured between the gate and source of the JFET. The IRS-B210 has a $R_g \approx 10^9\Omega$, making

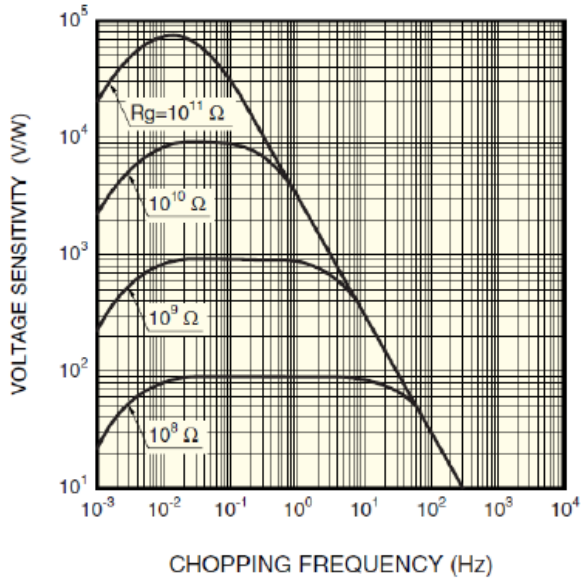


Figure A.4: Voltage sensitivity vs. copping frequency [43].

it most sensitive in the 10^{-1} Hz to 10 Hz range.

A.2 Experimentation

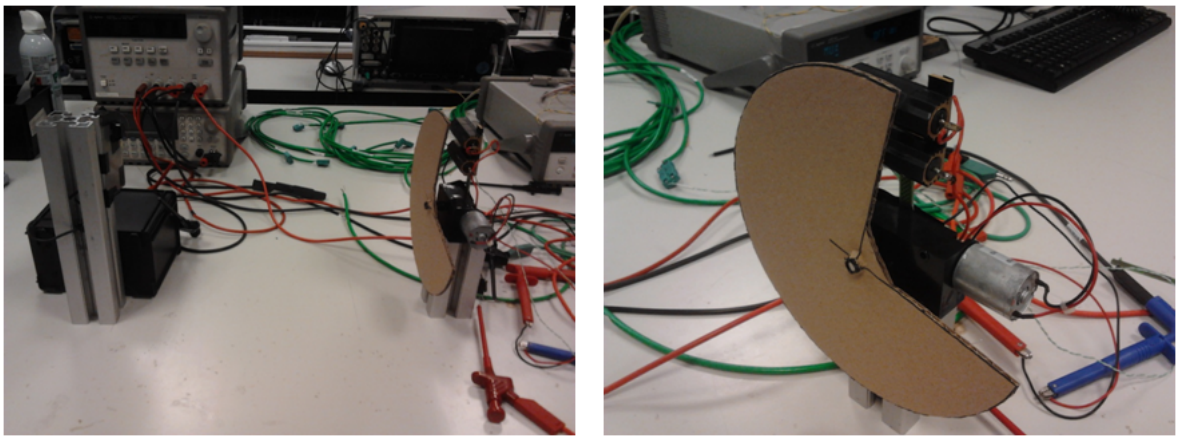
This section is divided into two parts. Section A.2.1 outlines the used test setup. This test setup is used for all experiments concerning the pyroelectric sensor, unless otherwise indicated. Section A.2.2 outlines the performed experiments.

A.2.1 Test setup

In order to qualify the sensor output for different speeds, distances and object sizes a mechanical chopper is used. A mechanical chopper consists of a disk which rotates with the aid of an electric motor. By removing a sector from the disk the FOV from the sensor is alternated between two scenes. The main advantage of a chopper is that it is used to simulate motion without physically moving the object itself. This allows for great reproducibility as the simulated motion only depends on the angular velocity for the electric motor. As $\omega \propto V$ for a brushed DC electric motor, different speeds can easily be simulated. This is something that is difficult with a live object, as generating a specific motion pattern is difficult and the speed between runs will vary.

Depending on the size of the window sector, the shape of the object and the angular velocity of the disk different function can be applied to the input of the sensor. Basic functions include: a saw-tooth function, a square wave function and a trapezoid function. For the analysis of the measurements a trapezoid function is assumed.

Figure A.5 shows an overview of the test setup used. On the left hand side of the figure the PIR sensor is shown mounted on a pole. Perpendicular to the sensor the object is placed behind the chopper. Figure A.5b shows a close-up of the chopper with, in this case, 2 objects



(a) (b)

Figure A.5: Test setup used; (a) overview of the entire setup with left the sensor and right the object, (b) closeup of the chopper with two objects mounted behind it.

are mounted behind it.

Unless otherwise indicated, for all experiments specially coated[1] power resistors are used as a substitute for actual persons. The main advantages of this are:

- The temperature of the object can be accurately regulated by regulating the power consumed by the resistor.
- The distance between the object and the sensor can be accurately determined, and easily kept constant.
- If needed the object can be completely stationary in the FOV of the sensor. Something which is nearly impossible with a live object, especially at close range.

These advantages allow for a greater reproducibility then when live objects, e.g. a hand, arm or complete person, are used.

Figure A.6 shows a close-up of the sensor. By limiting the FOV of the sensor to include only the object external disturbances are reduced. As there is no additional focusing done the amplitude of the output signal is not influenced by this. Besides reducing the FOV, also one of the two active elements blinded. As a result all experiments test the properties of a single element. An estimate of the chopper frequency as function of distance and speed of

[1]The resistors are coated using black PVC tape (CBBR7199 from Pro Power). The emissivity of PVC tape closely matches that of human skin, $\epsilon = 0.97$ v.s. $\epsilon = 0.98$ respectively [44].

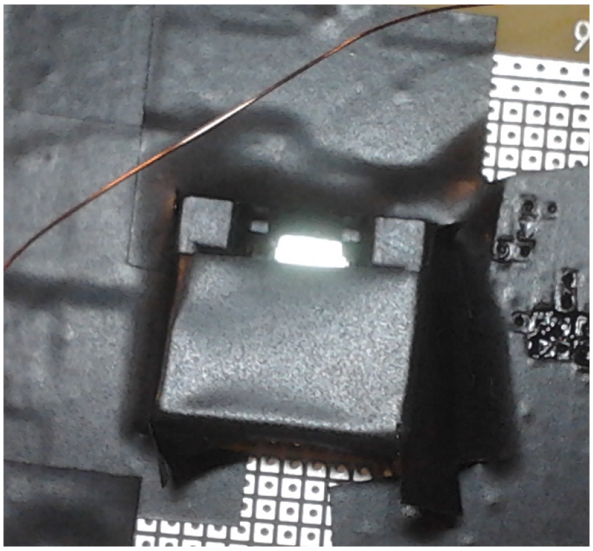


Figure A.6: PIR sensor with reduced FOV.

the object is given by the following equations:

$$l_p = 2d\tan(\frac{\gamma\pi}{360}) \tag{A.1}$$

$$\tau_{lp} = \frac{l_p}{v} \tag{A.2}$$

$$\tau_{disk} = \frac{360\tau_{lp}}{\theta} \tag{A.3}$$

$$f_{chopper} = \frac{1}{\tau_{disk}} = \frac{v\theta}{2d\tan(\frac{\gamma\pi}{360})} = \frac{v}{5.06d} \tag{A.4}$$

here l_p is the size of element projection on the object ([m]), v; is the desired simulated speed of object ([$^{\mathrm{m}}/_{\mathrm{s}}$]), d is the distance between object and sensor ([m]), γ is the FOV of the sensor ([°]), $\gamma = 70\,°$ for the used sensor [41]. θ; angle of window-sector of the chopper ([°]), $\theta = 90\,°$ in the test setup. τ_{lp} is the duration of object moving through the FOV ([s]), τ_{disk} is the rotation period of the chopper ([s]) and $f_{chopper}$; rotational speed of the chopper ([Hz]).

Equation A.4 only gives an estimation as some aspects are simplified. Most importantly the size of the actual element, which is simplified to be infinitely small, and the radial sensitivity, which is simplified to 1 for all angles. These simplifications are justifiable because: a) the object is placed perpendicular, both horizontally and vertically, to the sensor surface and the object location is stationary. This results in a constant sensitivity which is largest amongst all angels, and b) errors due to an incorrect l_p can be resolved by fitting the model to a waveform obtained from a sensor without a reduced FOV.

Filter and post-amplifier

For the measurements executed in Section A.2.2 a filter is connected between the output of the sensor and the measuring device. The goal of the filter is to both reduce background noise and increase the output signal amplitude. The implemented filter is a concatenation of

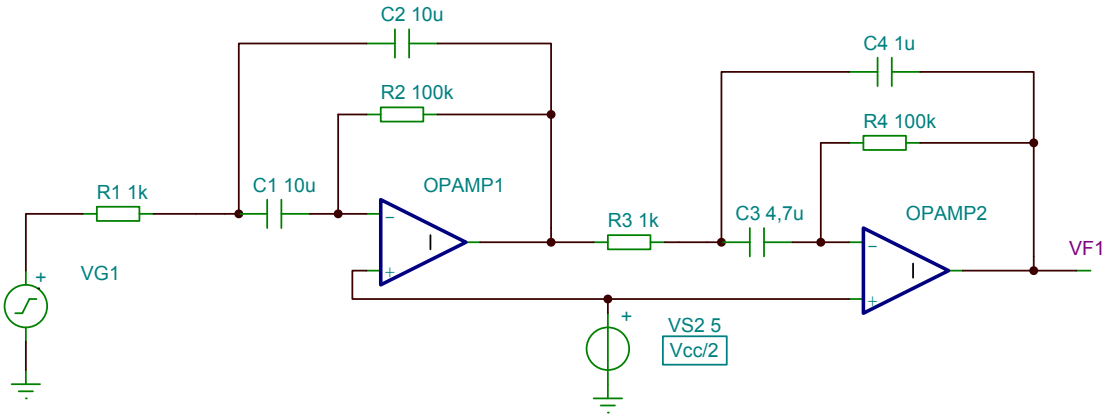


Figure A.7: Implemented band-pass filter

two high-Q band-pass filter with center frequencies of 1.6 Hz and 3.33 Hz respectively, and an overall gain of 50 dB in the pass band. The schematic of the filter is shown in Figure A.7. A high-Q band-pass filter is chosen instead of a regular band-pass filter as it is especially suited for narrow bandwidth high gain applications. Figure A.8 shows the frequency response of the implemented filter in blue. Red represents a regular band-pass filter (2-order Sallen-Key) with the same corner frequencies and gain. The difference in Q-factor is clearly visible, for example for a signal of 0.1 Hz the regular band-pass filter has a gain of 31 dB or 35×, whereas the high-Q filter has a gain of −15 dB or 0.18×. Note that both filter have the same roll-off rate for $\omega \lll \omega_{lc} \vee \omega \ggg \omega_{uc}$, where ω_{lc} is the lower cut-off frequency and ω_{uc} is the upper cut-off frequency. Which is $-40\,{}^{\mathrm{dB}}/_{\mathrm{decade}}$, as both are second-order filters.

A.2.2 Measurements

Initial measurements

The initial measurements are taken without a filter and in a completely shielded environment. The output signal is sampled at 2 Hz with an Agilent 34970A Data Acquisition/Switch Unit (DAQ). The DAQ has an accuracy of 0.5 μV and a resolution of 0.1 μV for VDC measurements and 0.03 °C and 0.01 °C respectively for temperature measurements [45].

Figure A.9 shows the output of the sensor as an object of 50 °C is moved into and out of view at a distance of 65 mm from the sensor[2]. The speed at which the movement accurse is not of interest in the figure, however the stead-state voltages are. After 103 samples (51.5 s) the object is moved into view, at sample index 332 (166 s) the object is moved out of view, this sequence is then repeated between sample index 572 (286 s) and 733 (366.5 s). This is also shown with the gray dashed line in the figure. It is shown in the figure that the steady state voltage is higher when the object is present then when the object is not present. For example at index 30 the output voltage is 0.571 mV, but at index 250, when the output has reached stead state, the output voltage is 0.577 mV. However the temperature sensor shows that the ambient temperature around the sensor is also increased by the presence of the object.

[2]Note that the peak at index 465 is a motion artifact generated by physical movement of the sensor.

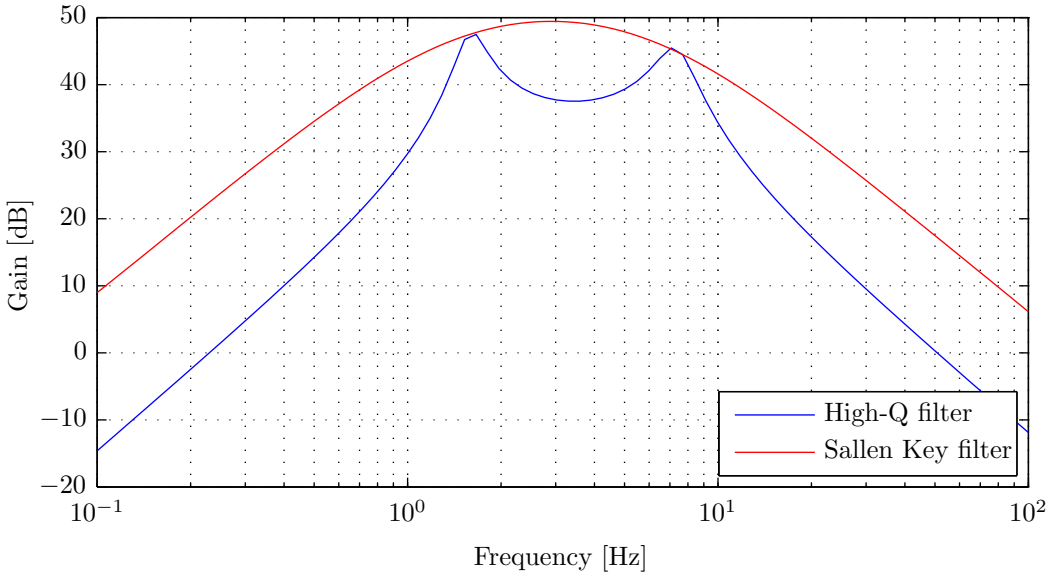


Figure A.8: Frequency response second-order band-pass filter.

The temperature is increased by 0.4 °C and 0.2 °C for the two different motions respectively. Figure A.9 suggests that the steady state temperature is dependent. As the thermal inertia of the temperature sensor is much larger than that of the PIR elements the thermal time constant is also larger, which explains why the sensor output is already in steady state while the temperature measured by the sensor is still increasing.

To confirm this the experiment is repeated with a lower object temperature. The result

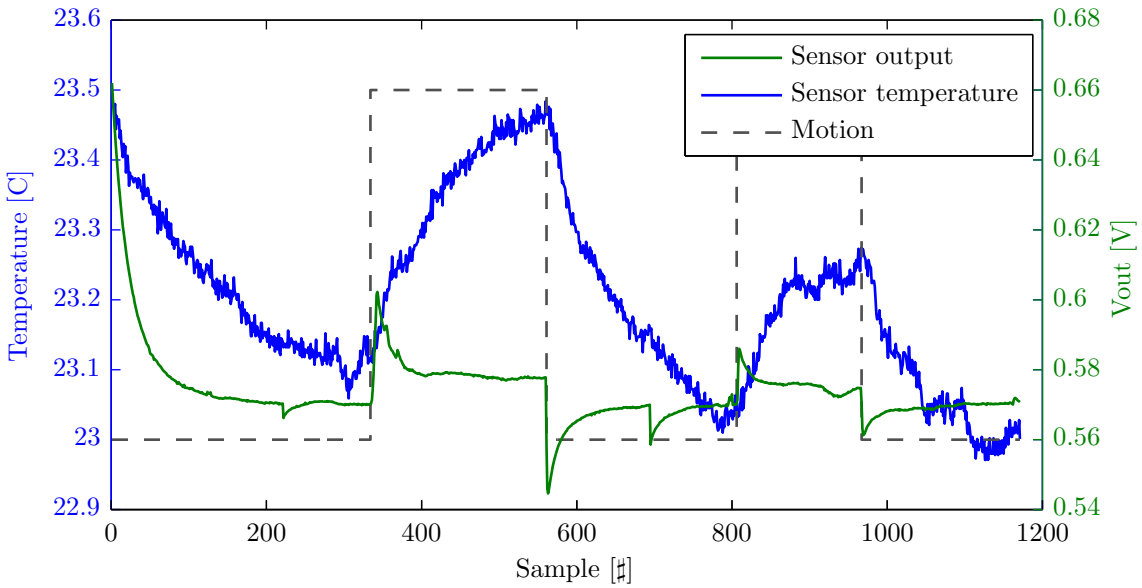


Figure A.9: Sensor temperature increase due to thermal radiation of object, $f = 2\,\text{Hz}$.

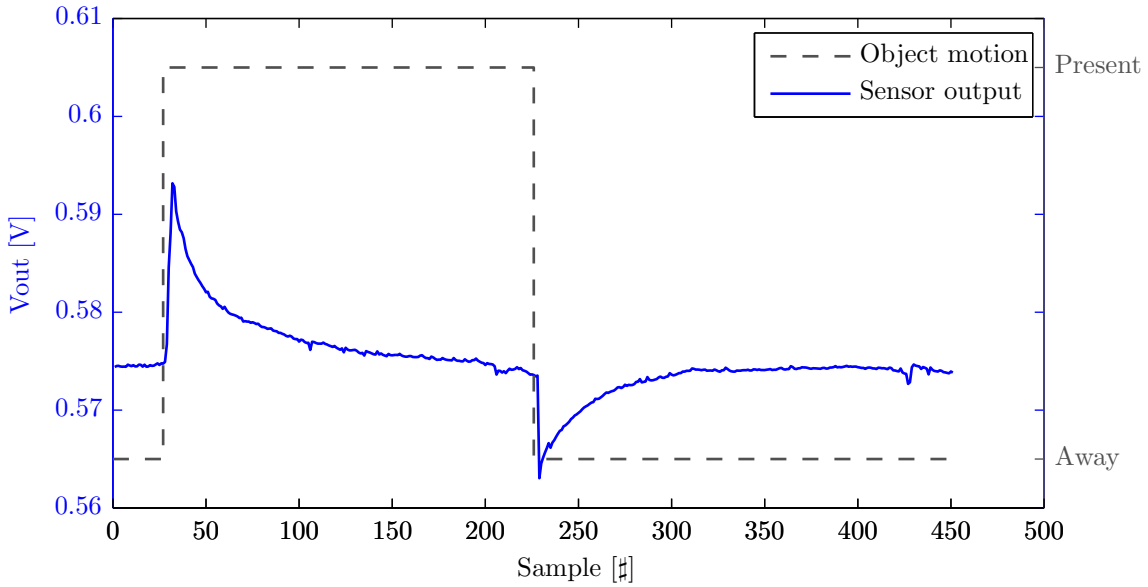


Figure A.10: Sensor output at constant sensor temperature, $f = 2\,\text{Hz}$.

is shown in Figure A.10. Here an object temperature of 30 °C is used, which results in a ΔT which is 4× as small. In this experiment the steady state voltage is independent, difference is less than 1 mV, of wether or not an object is present. The effects shown in Figure A.9 are also described in [22] p.81. When an object is shown to the sensor, first only the temperature of the outer layer is increased. However its temperature is almost instantly equal to the maximum temperature. Next with the thermal time constant the temperature spreads through the crystal dismissing the original thermal gradient that generated the charge and thus output voltage. However as the back side of the crystal is facing the ambient environment, which has an infinite thermal inertia, there will always remain a temperature gradient across the crystal. As a result of this temperature gradient across the crystal, or thermal flux through the crystal, the stead state voltage is increased with an offset proportional to this temperature gradient.

Figure A.11 shows the steady state output offset as a function of the temperature gradient. The offset can be calculated using the following formula:

$$V_{offset} \approx 14\,\text{mV} \cdot \Delta T \tag{A.5}$$

Figure A.12 shows the output of the sensor if the IR shielding around the test setup is removed. During these two experiments the sensor was facing a white wall and no other thermal objects where in its field of view. The object was placed at 50 cm from the sensor and the chopper was used to simulate a motion of $0.25\,{}^{\text{m}}/_{\text{s}}$. Compared to Figure A.10, which has a noise level of $0.2\,\mu$V, the noise level is much higher at more than 2 mV. It is in fact so high that in Figure A.12b it is difficult to determine which peaks are caused by motion and which by noise. They only differ by their frequency, which is much higher for motion then for background noise.

To reduce the noise and increase the signal the filter described in section A.2.1 is applied. A band-pass filter is used because from Figure A.9 and Figure A.10 it can be concluded that

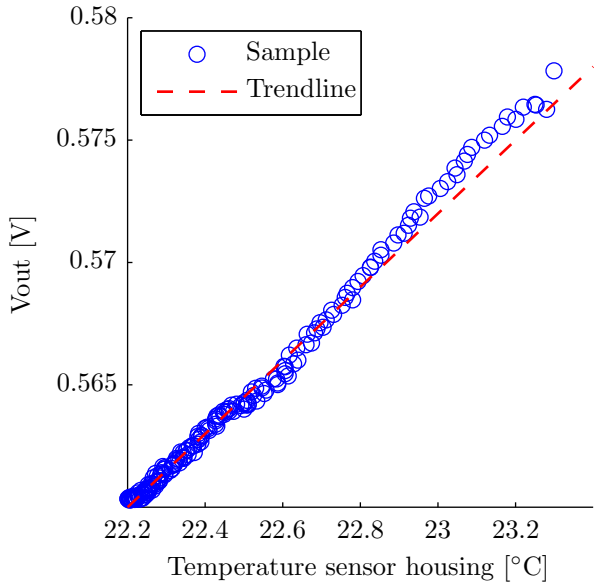


Figure A.11: Ambient temperature vs. steady state output voltage.

the DC component of the signal is of no practical use as it is too small. Even if a DC coupled amplifier is used to amplify this offset the required accuracy is too large to be practically viable.

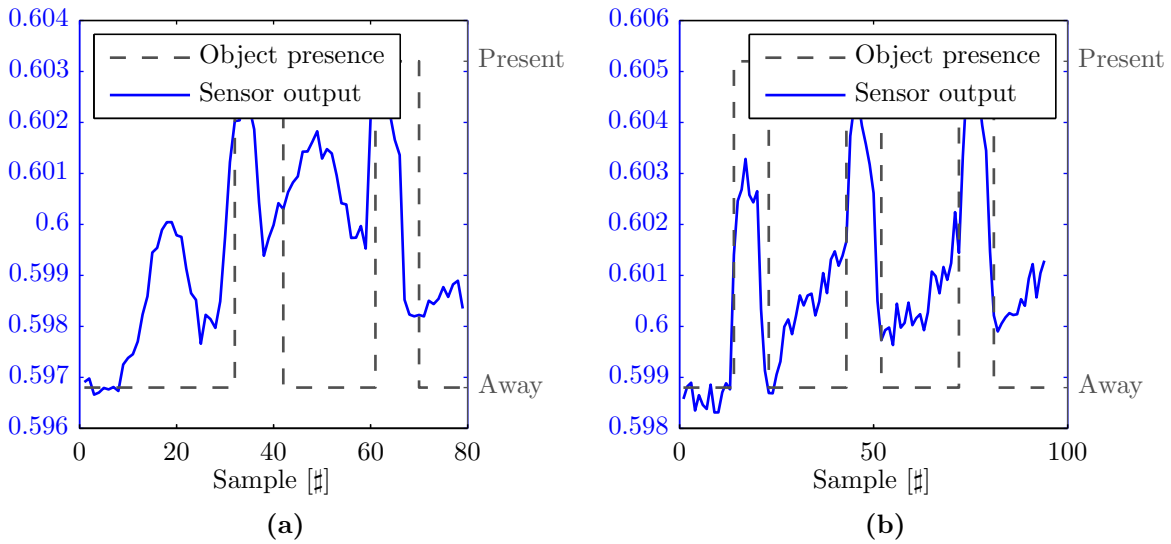


Figure A.12: Sensor output without shielding around test setup, $d = 50\,\text{cm}$, $v = 0.25\,{}^{\text{m}}/_{\text{s}}$, $f = 2\,\text{Hz}$.

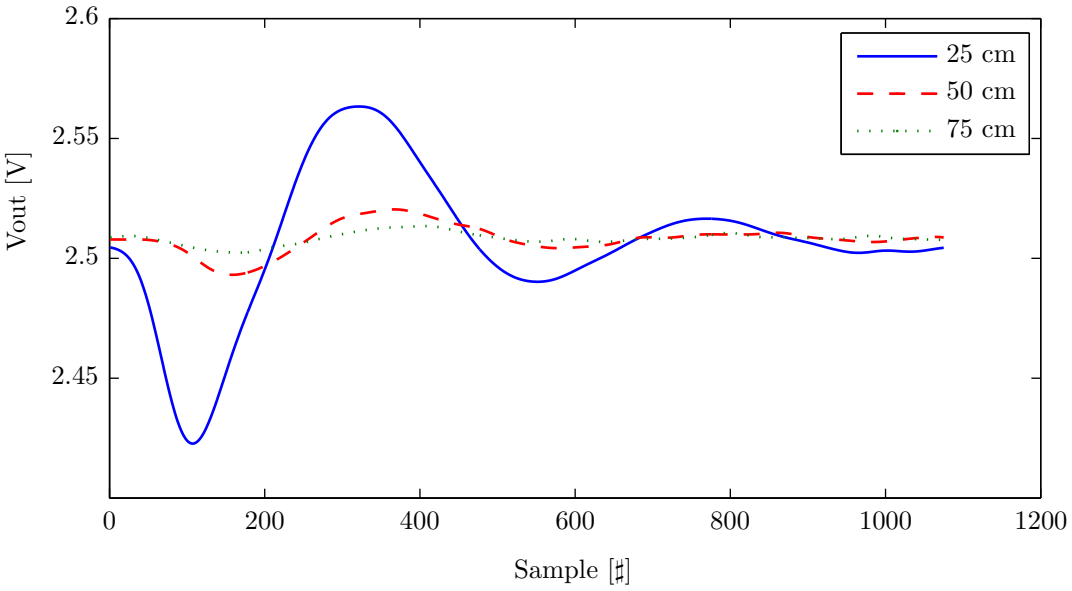


Figure A.13: Sensor output at different distances, $f = 500\,\text{Hz}$.

Table A.1: Maximum amplitude with respect to the distance

Distance (cm)	Min amplitude (mV)
25	83.22
50	15.07
70	5.93

Distance

During this experiment the object is placed at three different distances from the sensor. At each distance the object is moved at the same velocity through the field of view. For this experiment, and the ones described in section A.2.2 to A.2.2, the output signal is measured using a LeCroy WaveRunner 6100A[3] oscilloscope sampling at 500 Hz. Figure A.13 shows the result of this experiment. In this figure it shown that the distance greatly influences the amplitude of the output signal. The relation between the amplitude of the output signal and the distance to the object must be the inverse of the square. This is because the normalized intensity of IR radiation produced by the object varies inversely with the square of the distance, the temperature rise of the elements varies linearly with the absorbed IR radiation and the amplitude of the output signal varies also linearly with the temperature difference. Table A.1 shows the maximum amplitude with respect to the three tested distances.

Speed

During this experiment the object is placed at a constant distance of 25 cm from the sensor, its speed is varied between $0.25\,^{\text{m}}/_{\text{s}}$ and $1.5\,^{\text{m}}/_{\text{s}}$.

[3] http://teledynelecroy.com/

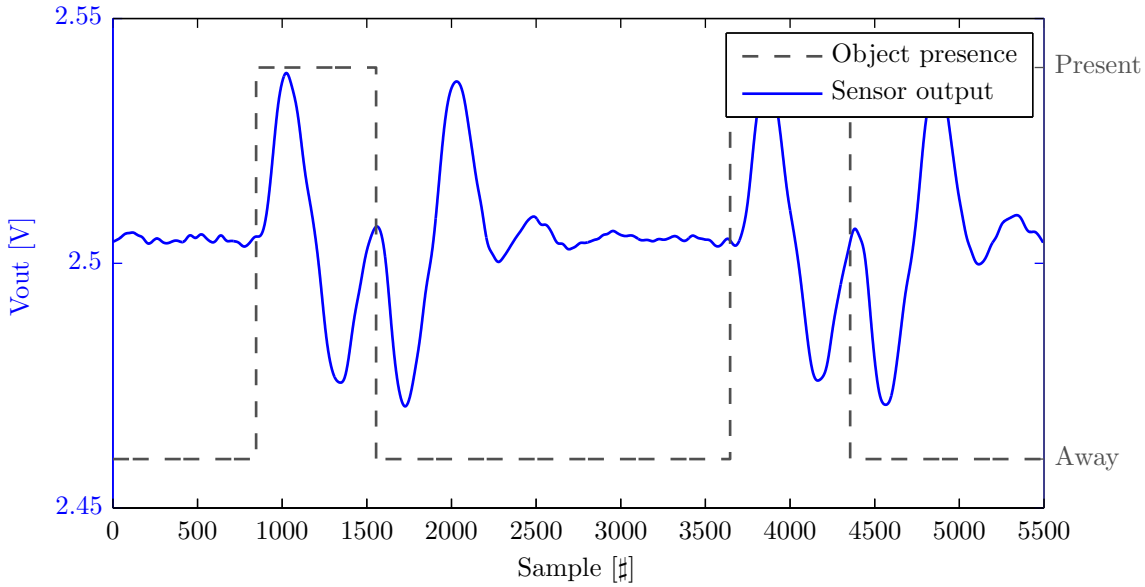


Figure A.14: Object moving; $d = 50\,\text{cm}, v = 0.25\,{}^{\text{m}}/_{\text{s}}, f = 500\,\text{Hz}$.

Figure A.14 shows the output of the sensor for an object continuously moving in and out of view at $0.25\,{}^{\text{m}}/_{\text{s}}$. The filter has not yet reached steady state after the object enters the field of view, before it leaves again. This is most likely the cause that the leaving pulses are smaller than the entering pulses. For this reason only single entering or leaving events are used in the creation of Figure A.15 and Figure A.16, elimination possible influences of entering events on subsequent leaving events. Figure A.15 shows the waveform for objects entering the field of view. Figure A.16 shows the waveforms for objects leaving the field of view. It is interesting to see that the frequency and the amplitude for the output signal is almost constant for speeds above $0.25\,{}^{\text{m}}/_{\text{s}}$. This is due to the used high-Q filter. This is also confirmed through simulation as shown in Figure A.17. The simulation shows the output for $v = 0.5\,{}^{\text{m}}/_{\text{s}}$ (green) and $v = 0.75\,{}^{\text{m}}/_{\text{s}}$ (red). It is shown that the output of the simulation is similar in amplitude and frequency to those obtained from the measurements. Figure A.18 shows that if a normal band-pass filter is used, the amplitude will become dependent on the velocity of the object. In this figure it is shown that the signal for objects leaving the field of view is inverted compared with objects entering the field of view. This is as expected as the direction of the pulse has a direct relation to the direction of the temperature gradient over the element. This gradient is positive as the object is placed into view and changes to negative when the object is removed from view.

The amplitude for objects exiting the field of view is on average similar in amplitude as for objects entering the field of view. The filter used in the simulation of Figure A.17 is the actual implemented filter shown in Figure A.7. The filter used for Figure A.18 is a standard band-pass filter with the same corner frequencies as Figure A.7, however without the ${}^{\text{vcc}}/_{2}$ offset generation and with a gain of $-14\,\text{dB}$. As a result only the shape of the waveforms can be compared between Figure A.17 and Figure A.18 and not their values.

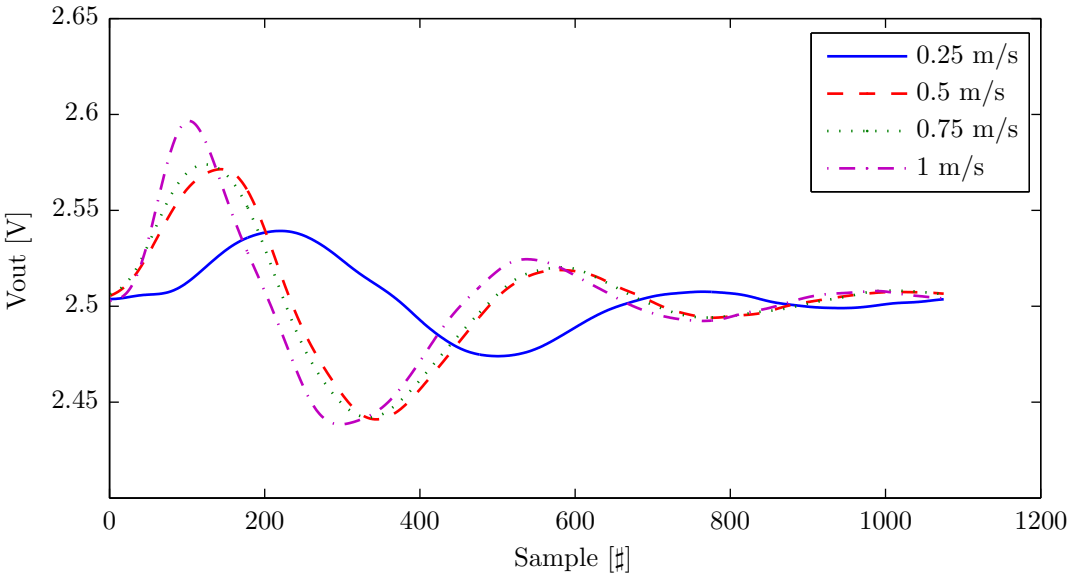


Figure A.15: Object entering view, $f = 500\,\text{Hz}$.

Multiple objects

During this experiment two objects are placed one after another in the view, subsequently one after another is removed from the view. The two objects are similar in size, however the first object is 58 °C whereas the second object is 93 °C. To compensate this, the first object is placed closer to the sensor and its velocity is increased. The thermal image of the test setup is shown in Figure A.19, here both objects are in view. Figure A.20 shows the derivative with

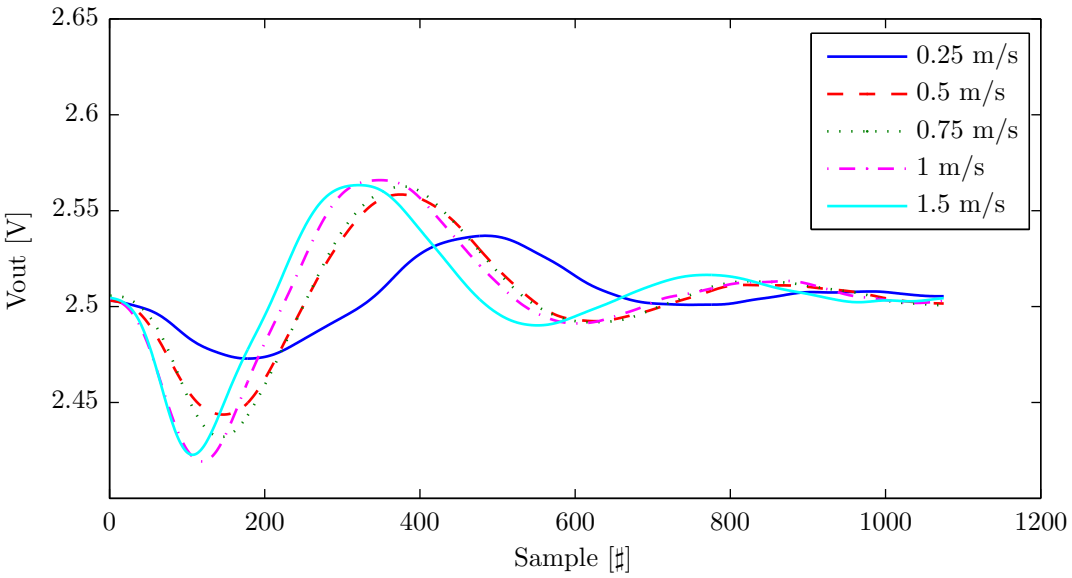


Figure A.16: Object exiting view, $f = 500\,\text{Hz}$.

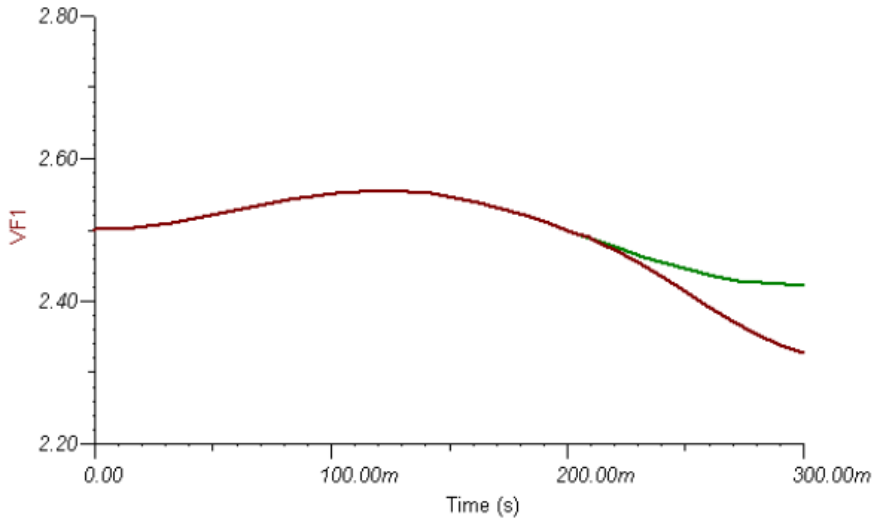


Figure A.17: Simulation output using high-Q filter for object entering view, $d = 50\,\text{cm}$, green $0.50\,{}^{\text{m}}/_{\text{s}}$, red $0.75\,{}^{\text{m}}/_{\text{s}}$.

respect to the temperature for when a) object one is entering the view and object two is b) entering and subsequently c) exiting the view.

Figure A.21 shows the sensor output. At time 2.5 s the first object enters the field of view, at time 7.5 s the second object enters the field of view. At time 12.5 s the second object leaves the field of view and at time 16.3 s the first object leaves the view.

The four actions are clearly visible in the sensor output. The amplitude of the second object is around 10 mV smaller than that of the first object, however this may also be caused

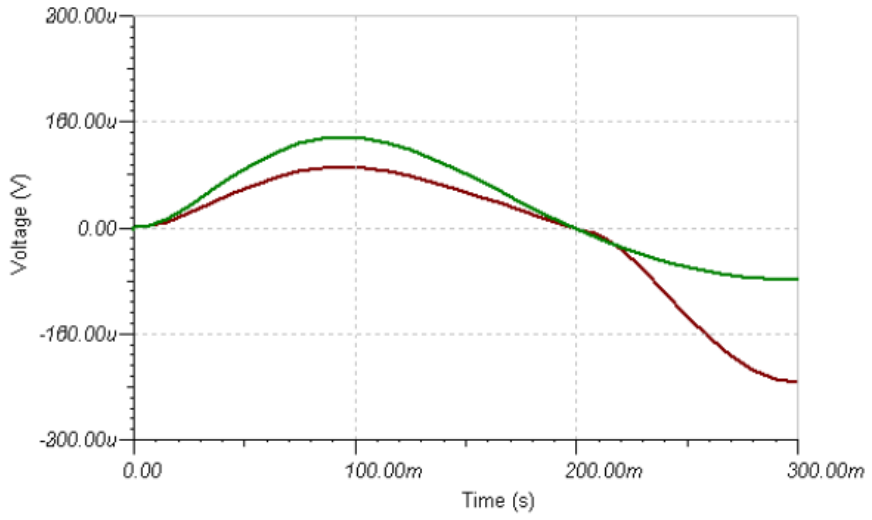


Figure A.18: Simulation output using normal filter for object entering view, $d = 50\,\text{cm}$, green $0.50\,{}^{\text{m}}/_{\text{s}}$, red $0.75\,{}^{\text{m}}/_{\text{s}}$.

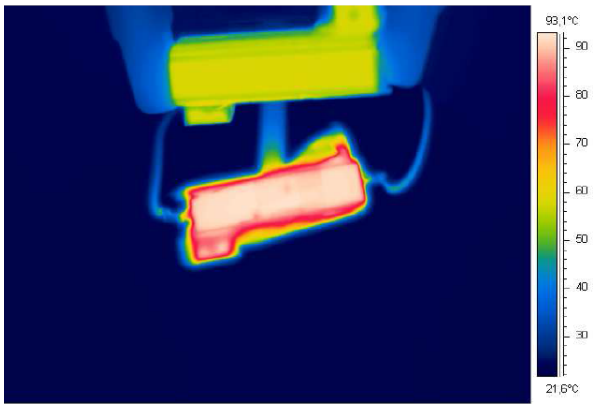
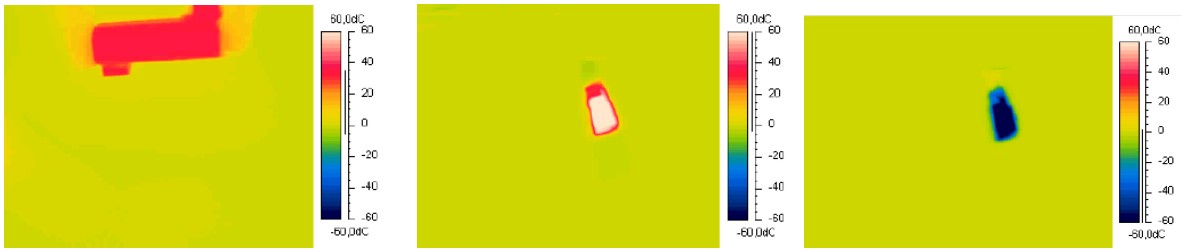


Figure A.19: Thermal image two objects in view.



(a) **(b)** **(c)**

Figure A.20: Derivative of the temperature when (a) object 1 is entering the view, (b) object 2 is entering the view and (c) object 2 is exiting the view.

by the difference in temperature between the objects. The polarity change between entering and exiting is clearly visible for both objects. Even though the speed differs over a factor of three between the objects, the frequency of the output signal is the same for both objects.

Scalability

The experiments performed in this chapter determine the scalability of experiments performed in chapter A.2.2 till A.2.2.

Figure A.22 shows the output of the sensor with respect to the object temperature. For these experiments the object is placed at a distance of 25 cm and a constant velocity of $0.25\,{}^{\mathrm{m}}/_{\mathrm{s}}$ is simulated using the chopper. Figure A.23 shows the same data, only as a scatter plot using the maximum amplitude of the sensor. In Figure A.23 it is shown that as the object temperature approaches the ambient temperature the amplitude of the signal becomes zero. Also included in the figure is a linear trend-line showing that the data is reasonably linear in the range 20 °C to 140 °C Figure A.23b shows a zoom of Figure A.23a on the 30 °C to 40 °C range. Figure A.24 shows the output for different object size placed at different distances. The purple line is a single object in view at 25 cm. For the green line the surface area of the object is doubled. For the blue line the area doubled and the distance is increase by a factor

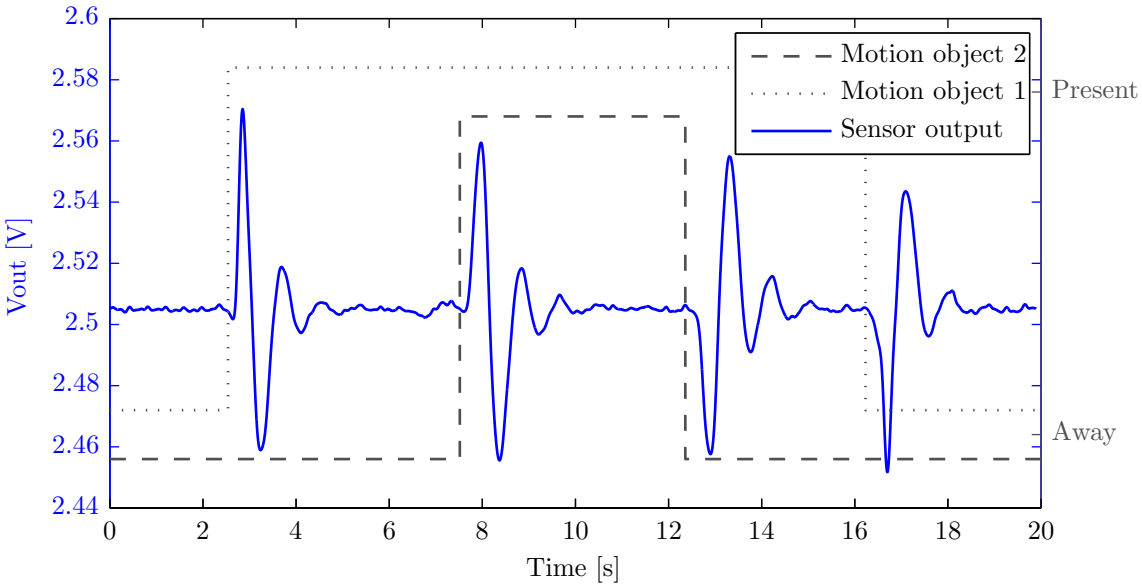


Figure A.21: Multiple objects in view.

of $\sqrt{2}$, which should as the result of the inverse square law result in the same output as single object at 25 cm. Two properties are shown in Figure A.24, namely:

- The amplitude of the signal is increased by 33% if the area of the object is doubled. This is counter intuitive, because if the area doubles the thermal flux through the sensor would also double. As the amplitude of the output signal has a linear relation with the thermal flux you would expect the amplitude to also double.

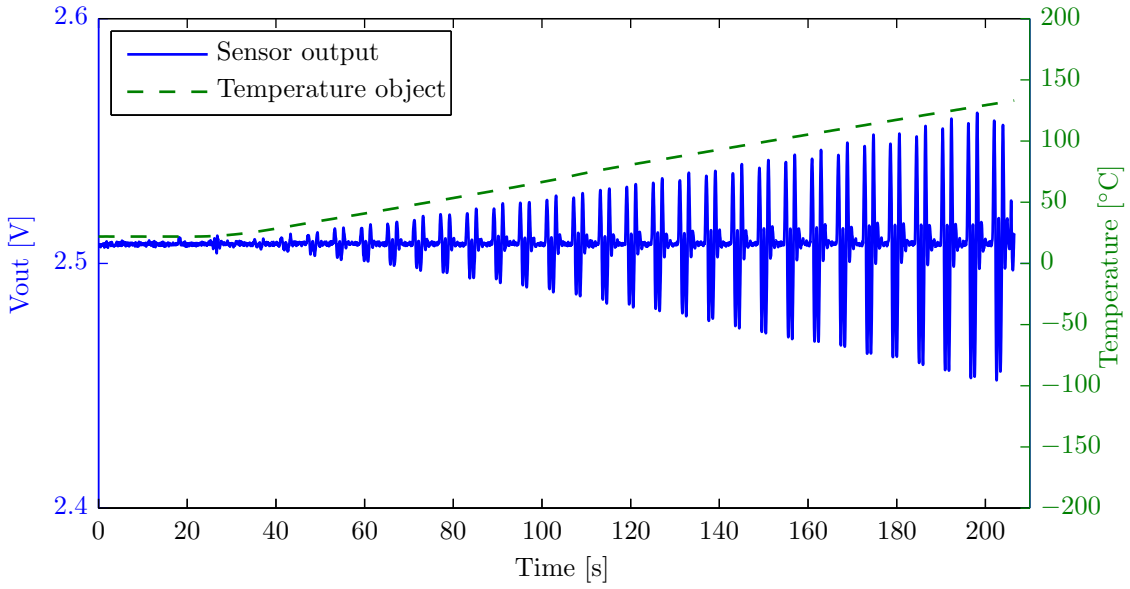


Figure A.22: Temperature response sensor.

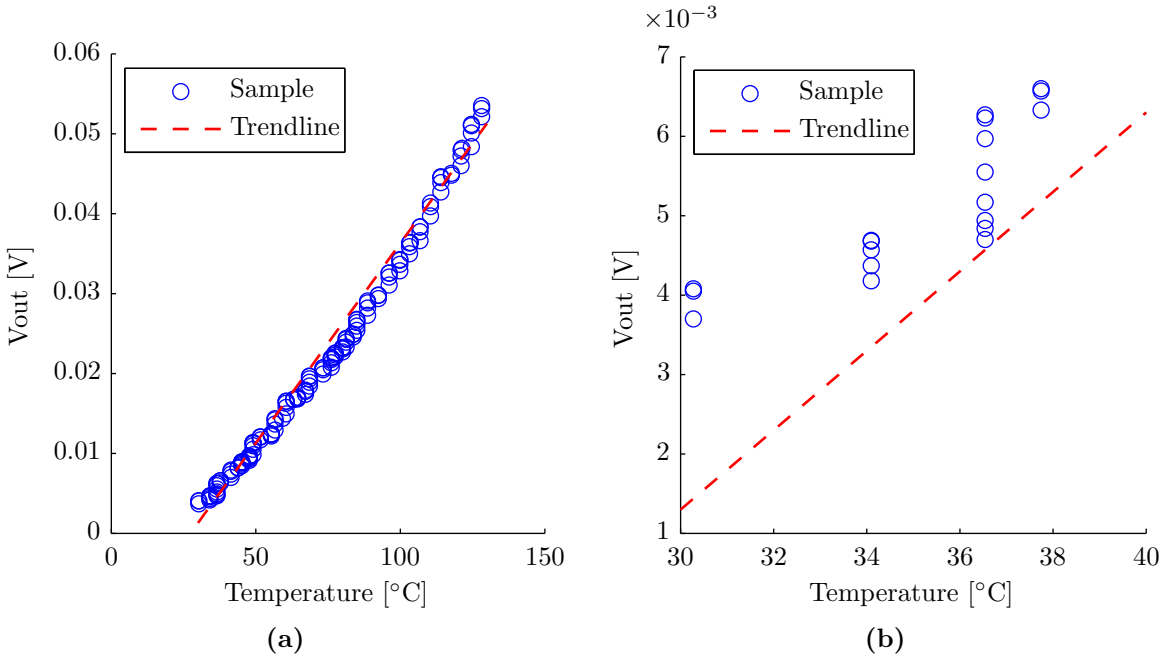


Figure A.23: Object temperature vs. output amplitude; (a) range $[0, 150]$, (b) range $[30, 40]$.

- The amplitude of the signal is reduced by 33% if the area is increase with a factor of 2 and the distance is increased with a factor of $\sqrt{2}$. This suggests that the amplitude is more than inversely proportional to the square of the distance to the object.

A.3 Conclusion

The experiments show that the amplitude of the output signal in combination with the filter is independent of the velocity, for velocities above $0.5\,{}^{\mathrm{m}}/_{\mathrm{s}}$. Furthermore it is shown that an already present object does not have any profound influence on the signal of a secondary object.

It is shown that the signal output increases linearly with the objects temperature, but more than inversely proportional to the square of the distance to the object. It is also shown that the amplitude scales less than linear with the size of the object. If the amplitude of the output signal is needed of the algorithm then its relation to distance and area must be further tests in order to detriment there relation more precisely.

From the experiments it is shown that the directions of the peaks can be used to determine if an object is entering or exiting and that the peaks only occur when an object is entering or exiting the field of view.

It is therefore shown that counting peaks in relation to their direction can be used to detect the presence of objects. However counting peaks has a few down sides which, at this point, cannot be solved using the PIR sensor alone. Some of these down sides are:

- Two objects entering simultaneously, but leaving separately. As the amplitude is a

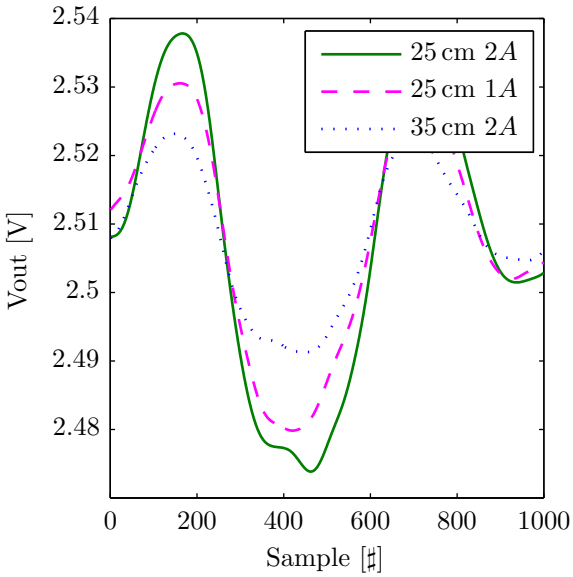


Figure A.24: Sensor output for different object sizes and distances.

function of the distance, the surface area and to some extent the velocity it is impossible to determine the surface area from the amplitude alone.

- Object moving very slowly. If an object moves slowly enough it will not be detected entering or leaving. If only the pulses are counted this immediately introduces an error, which in the case of an undetected leave cannot be corrected without additional assumptions.

- Two objects entering and leaving simultaneously. If two objects enter and exit simultaneously the output will be the sum of the two actions, which in best case allow for the detection of only one of the two actions.

The first and third point could be solved using the fact that the sensors are placed in a grid. For the first point, if to objects share their entire path together it has no negative effects on the algorithm. If they separate somewhere, their path could be traced back and the affected cells could be corrected. However backtracking becomes increasingly difficult if the size of the cells is increased. The third point could be solved by having the cells partially overlap. Compensating the simultaneous entering and exiting event of one cell with the entering and exiting event of two of its adjacent cells.

A possibility is to use a thermopile sensor in addition to the pyroelectric sensor. A thermopile sensor has an output which is directly proportional to the average amount of IR radiation it absorbs on its active surface. The output is thus the average temperature of the entire field of view. As a result it is only sensitive to the presence of objects and not their motion.

Three possible sensors are the SMTIR9902[4] of Smartec which has an analog output of about $2\,^{\mathrm{mV}}/_{\circ\mathrm{C}}$. The TMP006[5] of Texas Instruments which has a digital output with a reso-

[4] http://www.smartec.nl/pdf/DSSMTIR990X.PDF
[5] http://www.ti.com/lit/ds/symlink/tmp006.pdf

lution of 0.022 $^{\circ C}/_{bit}$. Or the GRID-EYE[6] from Panasonic, which has a digital output with a resolution of 0.25 $^{\circ C}/_{bit}$

[6] http://pewa.panasonic.com/assets/pcsd/catalog/grid-eye-catalog.pdf