Sensing Occupant Presence Using a Low Cost Thermal Detector Array

Chris Lelievre

February 2, 2017

This work was undertaken as a summer research scholarship funded by the Building Research Association of New Zealand and Victoria University of Wellington. Carried out by Chris Lelievre, a 3rd year student for a Bachelor of Engineering Electronics. Supervised by Manfred Plagmann and Ramesh Rayudu.

Abstract

As technology around smart homes grows and with the rapid expansion of the internet of things, being able to sense an occupant in a certain area is of growing importance. After researching different technologies to achieve this, it was decided to develop a device using a low resolution thermal sensor array. A device was built with a Panasonic Grid-Eye at its core and it managed to achieve approximately 95% accuracy sensing a single person.

Authors Note: Firstly to you the reader, thank you for picking up this paper. This project has been a really enjoyable and worthwhile endeavour for me. I deliberately tried to write this paper in such a way that anyone who was interested could pick it up and understand most of it so I hope you get something from it.

Secondly a big thanks to Manfred Plagmann, my industry supervisor at BRANZ for his support and giving me free reign over where this research went and to Ramesh Rayudu of VUW for his support from afar. A big thanks also to the other summer scholars at BRANZ for letting me bounce ideas off you, and to my fiancée Jessica for putting up with me.

Lastly I want to acknowledge Ash Tyndall and the authors of *Thermosense*, this paper and the device is very much built upon the work that they did so thanks to you again.

Contents

Note These links in the contents and references throughout this report are clickable (if you're viewing this electronically of course).

1 Introduction

With the current rise of the Internet of things (see 6.1.4) and developments around smart home technology, an expanding area of interest is being able to sense where people are. This project aims to explore the various technologies currently used for person sensing and develop a low-cost device capable of logging occupancy.

This project was undertaken as a summer research scholarship, jointly conducted by the Building Research Association of New Zealand (BRANZ) and Victoria University of Wellington (VUW). The desired outcomes of this project were firstly, to add to the pool of knowledge around indoor occupancy sensing, and secondly, to build a device that can sense whether someone is in a room, and possibly additional information about what they're doing.

This paper starts with a Literature Review, to determine what has already been done in the field of occupancy sensing and identify a research gap, followed by an analysis of the hardware and software used for this project's device and finally some of the results that were found and their importance.

2 Literature Review

A really good starting point for the field of occupancy sensing is the literature review by Teixara, Dublon and Savvides [15]. The focus of this review is not on classifying work that has already been done, but on identifying where more knowledge is needed, thus guiding the direction of this project.

2.1 Passive Infrared Sensors (PIRs)

2.1.1 PIRs alone

The most common way of sensing occupant presence is using PIRs. These are the kind of sensors found on automatic doors. They are very common because they're low cost and easy to use, however they are not very accurate and they can only tell if someone is there or not with no extra information.

A successful use of PIRs was demonstrated by Gunay et al. [8]. They used three PIRs in a single-door office and were able to determine with 95% accuracy if the office was occupied. A big limitation was whether the area covered by the PIRs included the door. This is fine in a research environment, but in business settings the options for where the sensors are positioned may be limited and so it might not always be possible to cover the doors. Also their 95% accuracy was achieved using short delay times (five minutes) and they were able to do this by optimising their decisions based on how these particular occupants behaved. This information is obviously not normally available. They believed a delay of 30 minutes would be more suitable for a real-world scenario, and with that parameter their accuracy dropped to 86% [8].

Other studies claim 98% accuracy using a single PIR and up another half a percent using multiple PIRs [9], though this was only over a two-minute period. They used decision trees for determining occupancy. Dodier et al. [5] used Bayesian belief networks to increase their accuracy also using three sensors. They found that if a their poorly located PIR was triggered, then it was only 20% probable that the room was occupied. However if that one and another better-located PIR fired then it was 98% likely that the room was occupied.

These studies show that PIRs are really useful for an immediate 'is the room occupied or not', however they are still limited in that as time progresses they cannot tell when someone is sitting still, what they are doing, or how many people are in the room. These limitations are things this project aims to solve, thus a solely PIR system would not be useful.

2.1.2 PIRs and other sensors

Interestingly, [9] found that while using decision trees, combining their PIR results with other sensors (CO2, acoustic) resulted in worse occupancy sensing. CO2 sensors are also known to be inherently slow as it takes time for a person's breathing to build up a significant enough amount of carbon dioxide. With that limitation and this study finding them unhelpful, it is unlikely they will be useful for this project.

Another interesting use of PIRs is using them for power savings, only turning on the main sensors after the PIR was triggered[9]. While a solely

PIR system would not be useful for our research, perhaps integrating them with another longer term sensor could be beneficial.

2.2 Thermal Imaging

2.2.1 Low Resolution

Low resolution thermal imaging is a promising technology for occupant sensing. Small sensors (around 64 pixels) are available for under $50(US) and have the potential to give more information than a PIR such as detecting multiple occupants, sensing motionless people and potentially even information about what they're doing.

For the Thermosense project [2], an 8x8 thermal sensor array was used. Feature vectors, various data mining algorithms and calibration of the thermal background were used to classify occupancy. Their best accuracy results were achieved using a k-nearest neighbours algorithm (see 6.1.2) with a root mean squared error of 0.346 (6.1.1 contains a quick explanation of RMSE).

For an honours project, Tyndall[16] used a 16x4 sensor and started by trying to replicate Thermosense's results. However, due to the difference in sensors, he was only able to get close to their k-nearest neighbour algorithm with an RMSE of 0.364. Tyndall went on to use some different algorithms, K* [4] and C4.5 (see 6.1.2), and got better results, up to an RMSE of 0.304 and 0.314 respectively. However he got very few results and only used the system in a lab situation, under very controlled conditions.

An interesting project [12] was done moving a column of 32 IR sensors 94 discrete steps to produce a 32x94 image. This thermal picture was then overlaid on a normal video camera which ran image processing to identify humans. By comparing the two pictures, they claim to improve the accuracy of the normal camera's person tracking. However, they didn't publish by how much the accuracy increased. Also one must assume that their rig was large in size and nowhere near consumer ready.

The inherent benefits of low resolution thermal imaging, along with the small but promising body of research around it, leads to it being a very promising candidate for this research project. However it may also be software intensive to be able to identify a person from such a low resolution 'image'.

2.2.2 Higher resolution

Having higher resolution thermal imaging could produce some more valuable information, the technology exists but the sensors/cameras are currently prohibitively expensive. Izumi [11] at SEI developed a 320x240 thermal camera but didn't test it too much beyond building it. They did however use it to classify a person's size which shows higher resolution could be useful for identifying people.

Hobbs [10] built a $10, 96 pixel thermal camera in 2001 but it was quite large and thus didn't catch on.

Gade et al. [6] used higher resolution thermal cameras to count the number of people in a section of the stands at sports games. This is interesting as it shows that with higher resolution you can get good counts for multiple people in a given area with minimal privacy concerns. Also their work was on real-world data which adds more credence to their research.

Schneider Electric are supposedly working on an advanced occupancy sensor using ULIS's 80x60 thermal sensor [3] [14]. This design is reported to be in product development and should be available sometime in 2017, a big company's interest does show that there is potential in the higher resolution technology.

It seems, as expected, that having higher resolution can lead to more accurate detection and being able to more easily detect multiple occupants [6], but the added expense and that a commercial company could soon release a product using this technology reduce its appeal.

2.3 Other interesting methods

2.3.1 Wireless

A novel way of person sensing was done using radio tomography [17]. A network of radio transmitters was set up and a connection from each transmitter to every other transmitter was established. This allowed them to take a baseline channel estimate and then measure any path loss caused by moving people. This experiment was done in a lab setting but recently the same authors finished crowd funding for a commercial system [18] where you plug ten transmitters into wall sockets around your house and it uses an app to tell you where people are in the house. This method is really exciting as it could potentially be set up anywhere and if your background can be determined accurately enough you can sense both static and moving people.

A similar project was done by researchers at the University of Maryland using a small network of wifi routers to measure time of flight and thus map out where occupants are in a house [19]. The research they did is really only a proof of concept and they outline many challenges including that it is really hard to track multiple people especially if they are changing the environment (moving around furniture etc), but they also have really promising results, claiming a 1.0 probability of detection and tracking someone's position with 86% accuracy.

While these technologies are really interesting it would be a steep learning curve to even implement what's already been done in these fields, let alone improve it.

2.3.2 Opportunistic sources

Researchers have also tried to determine where people are without using any extra hardware [7]. A study was done in a business setting using information from employees' calendars and their laptop location (determined by wifi) to track the employees. They claimed 90% accuracy which seems really high considering they're just using smart algorithms. However, it must be assumed that the accuracy will depend heavily on how accurately a user feeds the system: it's probable that some employees may forget to update their calendars or not take their laptops everywhere with them. Also it could not be used in a residential setting where people will move around multiple rooms quickly, rather than at work where people stay in the same place for long periods.

2.4 Conclusion - Research gap

From this research it was decided for this project to explore low resolution thermal imaging. It is slightly more expensive than PIRs but it is believed possible to get some more interesting data while still being at a low price point. Along with this are the possible benefits of sensing still people over time and potentially multiple people.

3 Design and Implementation

The core of this project is its smart software, trying to get the most out of the low-cost, low-quality hardware. But before delving into the software it makes sense to talk through the hardware first.

3.1 Hardware

For information on actually using the device and pictures, refer to the user documentation (included as an appendix to this report 6.2).

Panasonic Grid-Eye The Panasonic Grid-Eye AMG8832 is an 8 pixel by 8 pixel small thermal camera, rated to a range of 5m and available for around $40US. 64 pixels is actually quite small but it was an easy choice to use this sensor, mainly due to the lack of competition. Melexis make a similar sensor but its aspect ratio is 16x4 which is not very useful for our application. So even though we had trouble importing the Grid-Eye to New Zealand on its own (without a huge evaluation board), it was worth the trouble.

Feather board After starting coding on an arduino it was decided to port to a Feather M0 adatalogger due to the need for more RAM. Feather boards are adafruit's microcontroller break out boards as competition to Arduino. The board has been great, it's got a fair bit more power than Arduino boards, is much smaller (5 x 2cm), has a built in SD card slot (optional) and can be programmed through the Arduino IDE.
(If you're totally lost after reading that, read this quick guide 6.1.3.) The feather was an excellent choice for this project.

Cost One of the keys of this project was to keep the device as low cost as possible.

(See Table 1)

Table 1: Prices as at January 2017 (USD)

Grid Eye- Digikey breakout board	$50.00
Feather M0 adatalogger	$21.95
1200mAh LiPo battery	$9.95
JST connector	$1.46
Jiffy box	$1.09
Switch	$0.73
Piezo buzzer	$0.72
Diffused LED	$0.36
Total	**$86.26**

$86(US) isn't amazingly cheap but it's certainly cheaper than anything developed before (see Tyndall [16])

3.2 Demo Software

As a starting point, software was developed that simply read data from the grid eye evaluation board and sent it via a serial connection to a computer. This was useful to see what could be done with this data without the restrictions of a micro controller (having better speed, choice of programming language etc). This was harder than anticipated as Panasonic claimed that they provided libraries for doing this, but the link for these libraries on their website didn't work. Eventually a library on github was found to do the job [1]. Once this was working to a satisfactory level it was felt that the code was lightweight enough to be implemented on an arduino.

The software was first implemented on an Arduino Pro Mini (3.3V) which uses an 8MHz ATmega328. Due to the lack of restrictions on speed the low processing power was acceptable, however it did not have enough RAM to both save data to an SD card and run the recursive component checking algorithm at the same time. Thus it was decided to move to a Feather M0 Adatalogger as it has 32KB of RAM, 16x more than the pro mini, a faster processor (48MHz) and most importantly a built in SD card reader which was useful for keeping the final size of the device compact.

3.3 Final Software

This section describes how the software works. (See [13] for a link to a GitHub repository containing all this software)

3.3.1 Overview

The device first saves a snapshot of the sensing area to use as a thermal background. It also calculates the standard deviation across all pixels. It then compares each pixel's new temperature reading to its background and decides using the standard deviation whether the pixel is 'active'. It then looks for 'connected components' by checking whether an active pixel is adjacent to other active pixels. By determining the size of a connected component the device decides whether this is big enough to be a person and thus makes a decision about the number of people in the sensing area.

3.3.2 Connected Components

To determine whether the active pixels of a frame form a connected component both a wrapper and recursive method are used. First the list of active pixels is passed to the wrapper method which passes a pointer to that list to the recursive method. The recursive method takes the next value out of the active array, checks if there are any active pixels around it, and then calls itself again on the found pixel, returning the number connected pixels found each time.

Multiple People To determine if two people are in frame the connected components wrapper method saves the number of large enough components returned. If there are more than two of them the device decides there are two people in frame. (See 4.5 for results on how well this worked.)

3.3.3 Thermal Background

Method - Thermosense So far, the standard method of updating a low res thermal background was developed by the makers of Thermosense [2] and was also used by Tyndall [16]. It works by using a PIR (passive infrared sensor) to determine if any movement has happened in the last 15 minutes, if it hasn't it can simply merge the new background into the old with a much lower weight, if movement has been detected they use an exponentially weighted thermal average to merge a scaled version of the new background into the old with a much lower weighting.

However there are a few limitations of this method, firstly it relies on a PIR. For this project, it was hypothesised that similar results could be achieved without adding this extra sensor. Even though PIRs are cheap, it was felt that relying on a PIR again would not add anything new to the research around occupancy sensing which has already been done. Also this method reacts very slowly, 15 minutes is quite a long period and it was found that something simple like opening an external door or turning on a heater could drastically change the thermal background in a matter of a few seconds, and then with the low weighting of updating, the device could conceivably be wrong for several hours.

Algorithm 1 Update Background- Active Method

```
if (first time through) then
  background[pixel] = data[pixel]
else
  find active pixels
  if (more than 48 pixels active) then
    mean= mean + 2σ
    background[pixel]= background[pixel] + 2σ

    UpdateBackground {Recurse}
    return
  end if
  if pixel is not active then
    background[pix] = (background[pix]+data[pix]) / 2
  end if
  Calculate new standard deviation
end if
```

A New Method - Active To counter these issues a new method (henceforth referred to as the active method) was developed (Refer to Algorithm 1). The first time is pretty simple, when the device is turned on it saves the first reading as the thermal background, setting any unreasonable values to 20°C. It also calculates the standard deviation across all pixels (Thermosense saves the standard deviation for each pixel). Then it continues to take nine more readings. Each time it uses the previous standard deviation to find active pixels. If a pixel is not active it simply adds the previous background and the new reading together and divides by two. If the pixel is active it assumes that pixel must be occupied, so there is no need to update. It then calculates a completely new standard deviation.

A strength of this method is by not updating active pixels a person in the room should never become integrated into the thermal background. However, if something else hot is in the field of view it also won't be integrated into the thermal background unless it was there on start up, but it should not be picked up as a large enough connected component. Now because a person won't be integrated into the background, it works really well to update it more often, the device updates approximately every 20 seconds, and this makes it very resistant to noise and sudden temperature changes.

Active Method - Overheat trigger One problem with this method is that because we don't update hot pixels, if the whole room is suddenly heated up then all the pixels become active and don't get updated. To counter this, if the program recognises that most of the pixels are active (more than 49 out of the 64) it increases the background of every pixel as well as the mean by two standard deviations, then calls the update background method again. This means if the field of view heats up quickly it can deal with it promptly.

These algorithms for updating the background are tested in 4.3.

4 Results and Discussion

This section presents and discusses some of the testing that was done in developing this device.

4.1 Early prototyping

To first gauge how the sensor would react and whether it was able to sense multiple people, a test was set up in the lounge of an empty house. The Grid Eye sensor simply sent back its data to a computer, via an arduino, where a python script saved three feature vectors, the number of active pixels, the number of connected components and the size of the largest component. The aim of this test was to save this data and then analyse it with Weka (see 6.1.2 for an explanation of Weka) to find a suitable algorithm which could be installed on an arduino or other microcontroller to decide in real time whether the space is occupied or not.

Over three tests, to improve accuracy, six people entered the frame, one at a time, and stood either 1.8 or 2.8m horizontally away from the sensor, which was mounted 2.2m up. Then adding ground truth of 0-6 people occupying the space, the Weka results were as follows:

Table 2: First results

Test	Algorithm	RMSE	Correctly Classified (%)
1	C4.5	0.2763	59.79
	knn	0.278	59.27
	k*	0.281	57.52
2	C4.5	0.302	50.51
	knn	0.278	59.27
3	C4.5	0.322	41.65
	knn	0.330	40.22

(see 6.1.2 for explanations of these algorithms and 6.1.1 for a quick guide on how to read these results)

The results for these first tests were quite poor. However it was learnt that,

1) Unsurprisingly, the accuracy is correlated with how accurately the sensor is picking up people. Just from looking at the active pixels map in real time it was obvious that the sensor wasn't always detecting people, and,

2) C4.5 was the most accurate (for correct classifications) out of the tested data mining algorithms. This is extremely handy as C4.5 builds a decision tree which can easily be implemented on microcontroller hardware.

A positive result was that the accuracy greatly improved when all these tests and a few more were put together and analysed for different categories. When trying to determine whether 0, 1, 2 or more than 3 people were in frame, C4.5's accuracy increased to 67.62%. Then for categories of occupied or not the accuracy was 84.42%. This is very good because the Grid Eye's sensing area is quite small so it would be monumental to be able to get it recognising six individuals at a time, but if it is very accurate with 'is there one person or not' then that is useful.

It is worth mentioning that in these tests knn was very slightly (less than half a percent) better at classifying the images however due to the complicated nature of knn it is still much easier, without sacrificing very much accuracy, to use a decision tree.

Table 3: Thermal backgrounds

Set	Occupant	Conditions	Accuracy Active Method (%)	Accuracy Thermosense (%)
1	-	Normal Temperature	100.00%	100.00%
2	Sitting	Normal Temperature	97.13%	97.47%
3	Standing	Normal Temperature	100.00%	100.00%
4	-	Normal Temperature	97.31%	0.00%
5	-	Cold wind	96.85%	77.07%
6	Sitting	Cold wind	83.98%	60.27%
7	-	Cold wind	65.25%	0.00%
8	-	No wind	10.90%	1.73%

4.2 Optimising

When classifying active pixels, Tyndall [16] and Thermosense [2] used more than three standard deviations (σ) above the mean as criteria for a pixel to be active. For the Grid Eye, this was not sensitive enough and it was found that two σ out led to better accuracy.

It was then tested whether it was better to average the new temperature value with the last four values or with just one value. With the four values, the raw value is saved, meaning each value was weighted by 0.2 when calculating the current temperature. However for two values the saved value was-

$$temperature = \frac{newvalue + lastvalue}{2}$$

Which is the same as (v is value, t is time)

$$Temperature = 0.5v(t) + 0.25v(t-1) + 0.125v(t-2) + 0.0625v(t-3)...etc$$

meaning that very old values still have a small impact on the current result. This method was found to be slightly better, 92.41% correct vs 91.09% correct, and due to its ease of implementation, the averaging two values was kept.

Through further testing it was also found that having airflow through the sensing area, the room being hotter than approximately 25°C and the occupants wearing sweatshirts all severely limited the accuracy of the sensor.

For subsequent tests, with and without a through breeze, accuracy improved from 79.46% with the wind, to 92.41% without. Table 2 was in a hot environment, where the Grid Eye's mean was around 24°C, in an air conditioned environment, where its mean read 21°C it achieved 98.17% accuracy with the same set up. And finally in another test with the occupant wearing and then removing their sweatshirt accuracy jumped from 55% to 78%.

It was also found that having the sensor closer to the occupants was, unsurprisingly, beneficial. Going from having the sensor approximately 2.7m from the occupants face to 2.55m increased accuracy from 82% up to 92%. The Grid eye is rated to have a range of 5m but these experiments showing that getting within 2.5m greatly increases your chances of being able to distinguish a person from noise. From this result it was decided to mount the sensor to the ceiling, rather than having it horizontal (on a wall), as this usually decreases the distance to the occupant.

4.3 Comparing Thermal Background Methods

An important test was done to compare both Thermosense's [2] and the new active method (Algorithm 1) of updating the thermal background. This test was done on a day with very hot sunlight but a strong cold wind. Thus, the temperature of the room could be changed very quickly by opening the door, which let the wind in cooling down the room, or leaving the door closed and the room heated up.

Just a note, this test does not reflect on Thermosense's method in itself as what was tested had significant differences (to their original implementation), the test was to ascertain what would happen when you try to use the algorithm without a PIR and updating much more often (about every 20 seconds rather than every 15 minutes).

(See Table 3)

The most telling result is at set 5 where the cold wind is introduced so the background temperature goes very quickly from hot to cold. We can see that the Thermosense method performs much poorer than the active method, likely due to its low weighting of the new background when updating. Also, interestingly, the Thermosense method bottoms out when going from occupied to unoccupied. This could be a bug in the code or it could just be that the method is poor, either way due to the better reaction in changing temperature it was decided to use the active method. However, both methods don't handle going from cold to hot well (in set 8).

4.4 Overheat trigger

To counter the poor response when going from cold to hot, an overheat trigger was incorporated into the code. You can see it in Algorithm 1 after the statement

if (more than 48 pixels active) **then**
Much testing was done to optimise this trigger and in Table 4 the final result of the trigger is tested against calling update again after triggering when a room is heated up quickly.

Table 4: Overheat trigger

Heater	Update Once	Update Recursively
ON	8.13%	25.92%
OFF	89.07%	93.81%

From this table (4) we can see that the Overheat trigger has markedly improved performance going from cold to hot (10.90% up to 93.81%) (Tables 3 and 4). Also we see that updating recursively brings a slight increase in performance - hence why it is done in the final code (Algorithm 1). It is also worth noticing that the device reacted poorly while the heater was on. Wind definitely affects the sensor and the heater did blow out some hot air, but it is believed that while the room is heating quickly the sensor will also be inaccurate. Solving this problem will require further research.

4.5 Multiple People

Testing was done with the final device to determine how well counting the connected components worked for sensing up to two people. It was hypothesised that any more than that (two people) would be indistinguishable for the Grid Eye sensor.

Table 5: Multiple people

Occupants	Exact Classification	General Classification
1	79.18%	95.90%
2	91.44%	100%
2 standing close together	91.44%	100%
2 stepped into frame together	45.99%	95.85%
2 stepped apart	98.33%	100%
2 back together	39.35%	100%

In Table 5 the general classification column counts both two people and one person classifications as correct if the space is occupied. The exact classification column is if it is in the right category.

From this experiment we see that when two people are standing close together it has difficulty distinguishing them as two separate entities, but it does know whether the space is occupied or not.

These are good results as Tyndall's ([16]) best result is 82.56% classified correctly and that was using a K* algorithm on a computer.

5 Conclusions

This project began by choosing a technology to investigate, with the goal of increasing the knowledge of occupancy sensing. After settling on low resolution thermal imaging, a device was developed and tests were carried out to determine what could be done.

It was found that taking the active pixels into account when updating the thermal background offered improvements to robustness in changing environments. Also that the sensor was accurate enough to detect multiple people, but its field of view was not wide enough for this feature to be particularly useful. The device worked with approximately 95% accuracy and that 5% is likely due to the shortness of the tests. It is not believed that this device could be any more accurate in normal conditions.

It is hoped that this work will be useful not only to BRANZ in their research but also to others working in occupancy sensing.

6 Appendices

6.1 Acquired Knowledge

This section details some useful information learned over the course of this project and attempts to explain or clarify some terms and ideas used throughout this paper.

6.1.1 Correct Classification vs RMSE

This is a little guide on how to read this paper's results and will hopefully be helpful to you, the reader, in understanding what those results mean.

Usually in occupancy-based projects, people measure their accuracy using Root Mean Squared Error (RMSE). This is the measure of how close you are to the correct answer and is used due to its applications to machine learning. When you read a RMSE value it is in the units that the original value is in, so if there is an RMSE of 0.3, most of the time in these papers that means it's off by ± 0.3 persons. Personally, I find RMSE unintuitive and not particularly suited to the discrete categories being measured here. For that reason, all results in this paper measure accuracy with a Correct Classification percentage. So if 'not occupied' is expected, the % value is

$$= \frac{NotOccupied}{Occupied}$$

and if 'occupied' is expected the fraction is flipped. So the Correct Classification percentage is how often the device is right which is hopefully more intuitive. You can see RMSE and percentage side by side in Table 2 because those results concern machine learning algorithms. From then on RMSE is not included.

6.1.2 Data Mining

Data mining is a statistics field used to generate useful information or make predictions from a set of data. Each entry involves some information called feature vectors and the ground truth. From this an algorithm would usually look for patterns in the feature vectors to learn how to predict something given a set of data but NOT the ground truth. This is useful in this project as it is wanted for a PC to look at a set of example data, determine a pattern or method for predicting occupancy, then that method can be implemented in software built in to a small device.

Weka Weka is a data mining program developed by the University of Waikato. It was useful to this project due to its long list of already implemented algorithms which can easily be applied to most datasets.

Algorithms These algorithms are the ones used most successfully by [16] [2] in their similar projects involving Low Resolution Thermal Arrays.

C4.5 This algorithm builds a decision tree from the data set. The strength of decision trees is that they are very simple. If you remember those quizzes that you used to get in magazines that had a large upside down 'tree' of questions, then depending on your answer you go to a different question and at the end it tells you something like which superhero you are, that's a decision tree. For a programmer a decision tree is a collection of if/else statements making it very easy to code. Its main weakness is that it can tend to over-fit to its training dataset.

KNN and K* [4] These are both 'Lazy' algorithms which means they don't do any computing until classifying instances. K* (K-star)[4] is a variation on KNN which was of interest due to Tyndall[16] finding it the most accurate way of classifying frames.

6.1.3 Microcontrollers

A microcontroller is a cheap, small, dumb computer. They're easy to program with a small set of instructions and basically every piece of semi-complicated electronics you see will have one or more inside providing its brains.

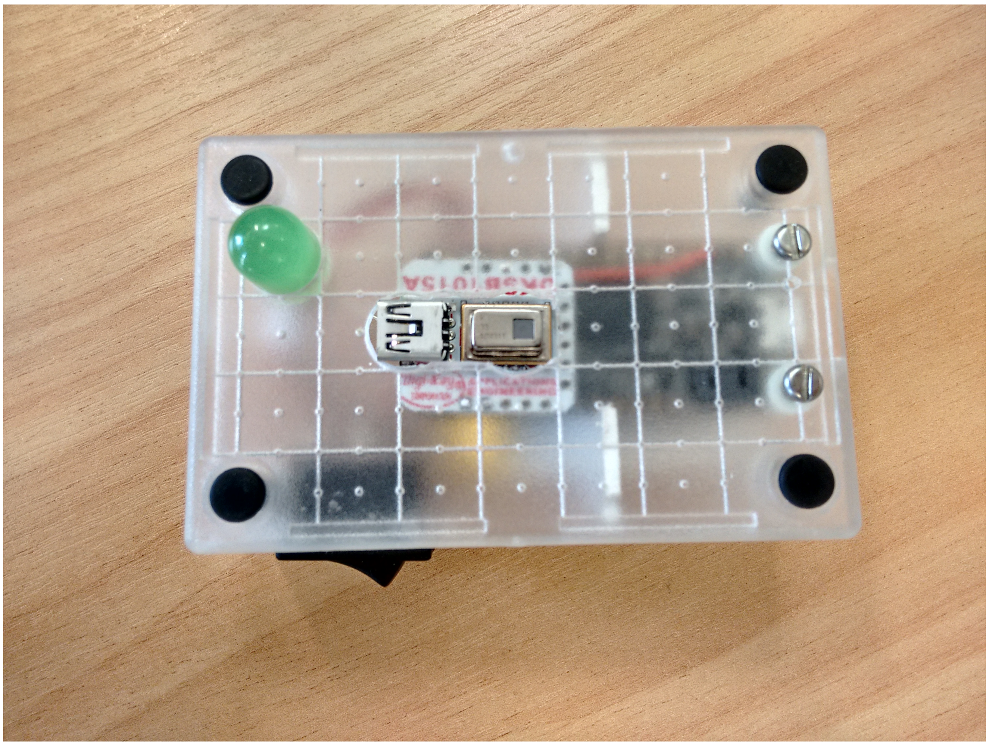
6.1.4 The Internet of Things

The internet of things is the term given to the technology trend of connecting as many devices to the internet as possible. The goal being more automation in our lives. For instance when your car knows you are close to home it might turn on the heating. Or when your alarm clock goes off in the morning it could turn on the coffee machine and toaster.

6.2 User Documentation

BRANZ Occupancy Sensor System
B.O.S.S.
Technical Guide
Chris Lelievre

February 2, 2017



6.2.1 Introduction

This document is meant to help and introduce you to the BRANZ Occupancy Sensor developed as part of a summer scholarship in the summer of 2016/17. In this guide you'll find how to build a unit yourself, how to use the provided GUI and the expected performance of this device.

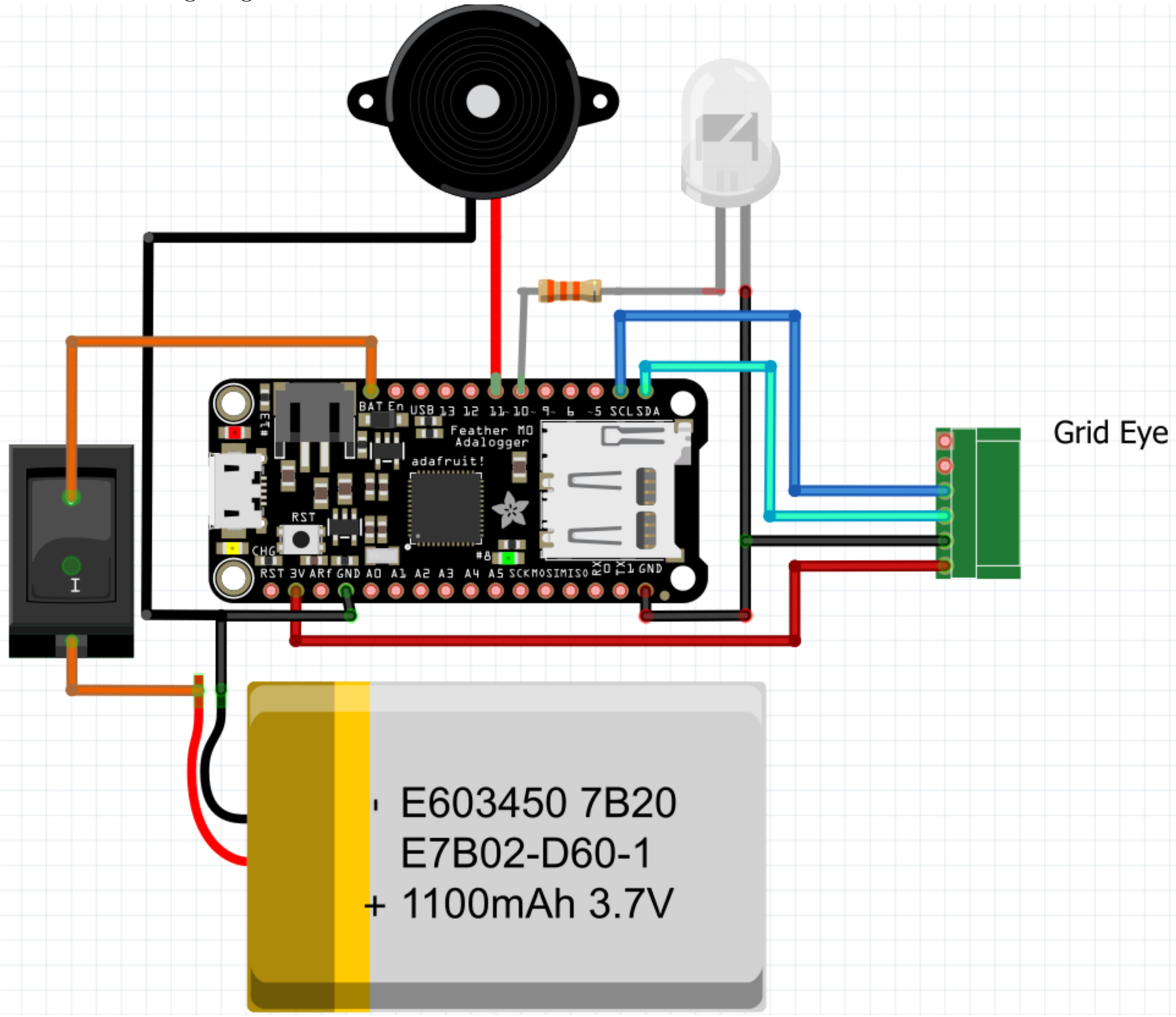
6.2.2 Build a Unit

The non-generic parts you'll need are a Panasonic Grid-Eye, I used an AMG8831 though I believe it should work with other models. To save time I brought a breakout board from Digi-Key.
`http://www.digikey.com/product-detail/en/digi-key-evaluation-boards/DKSB1015A/906-1002-ND/4360804`
Feel free to make your own PCB, all you need is access to the I2C pins (Pins 2 and 3 on the Grid Eye).

A Feather M0 Adatalogger
`https://www.adafruit.com/product/2796`
The Adatalogger is used due to its built in SD card reader.

Other than that you'll need a Lithium-Polymer battery, a switch and a piezo buzzer. I also included an extra LED (with 33kΩ resistor) in mine but that was mostly for debugging.

Below is a wiring diagram of the device



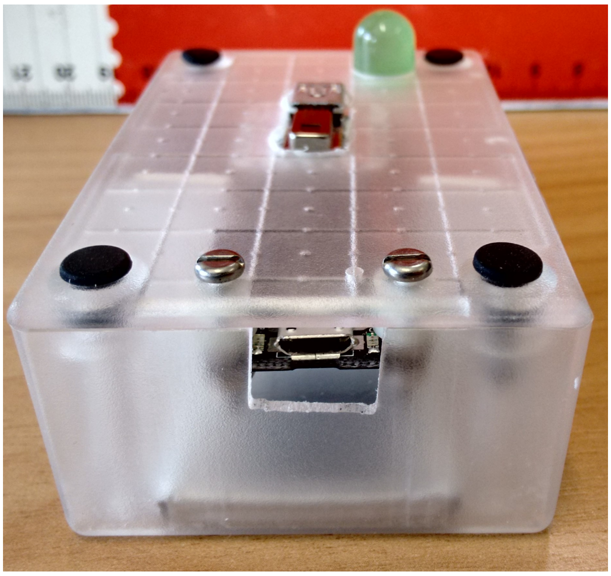
6.2.3 Using the Device

Positioning Mount the device to the roof above the area you want to monitor. I find it easiest to use Blu-Tack. Best results are obtained when the sensor is within about 2m of the occupant, of course closer is better.

Turning it on Each time the device is turned on it will create a new file on the SD card. It also starts by collecting a thermal background so it the sensing area should be empty between turning it on and the device finishing beeping. It will beep 10 times followed by a happy tone on start up. It is often useful to note down the time you turned the device on so that the clock values can be adjusted later from the accompanying software.

The Green Feedback LED The large green LED on the front tells you when the space is occupied or not and is helpful to tell if the device is measuring correctly or not.

Charging To charge the device plug it into a computer using the micro USB port on the feather (broken out on the original device)



6.2.4 Windows Computer Software

When the device is connected to the computer, not only does it charge the battery (you will need to have the on/off switch on for the battery to charge) but you can save and delete files from the SD card.

Find COM The first thing to do is to find your device, click find COM. If there is only one COM port available the program will automatically choose that port (notified through the status bar). Otherwise click on the COM port your device is on. If you're not sure which one is your device, open the device manager (this can be done by right clicking on the windows button at the bottom left of your screen) and under "Ports (COM and LPT)" look for the Adafruit Feather M0. Then select this COM port back in the BRANZ Occupancy Sensor software.

Load The load button allows you to load in an occupancy log from the SD card. The status field shows progress and the green feedback LED blinks as it sends.
Once the file is loaded it can be read in the text pane and a graph is displayed.

Save A loaded file can be saved by choosing the save option. Otherwise the file can be found in the folder where the software was launched from. When saving it will take a bit longer than usual.

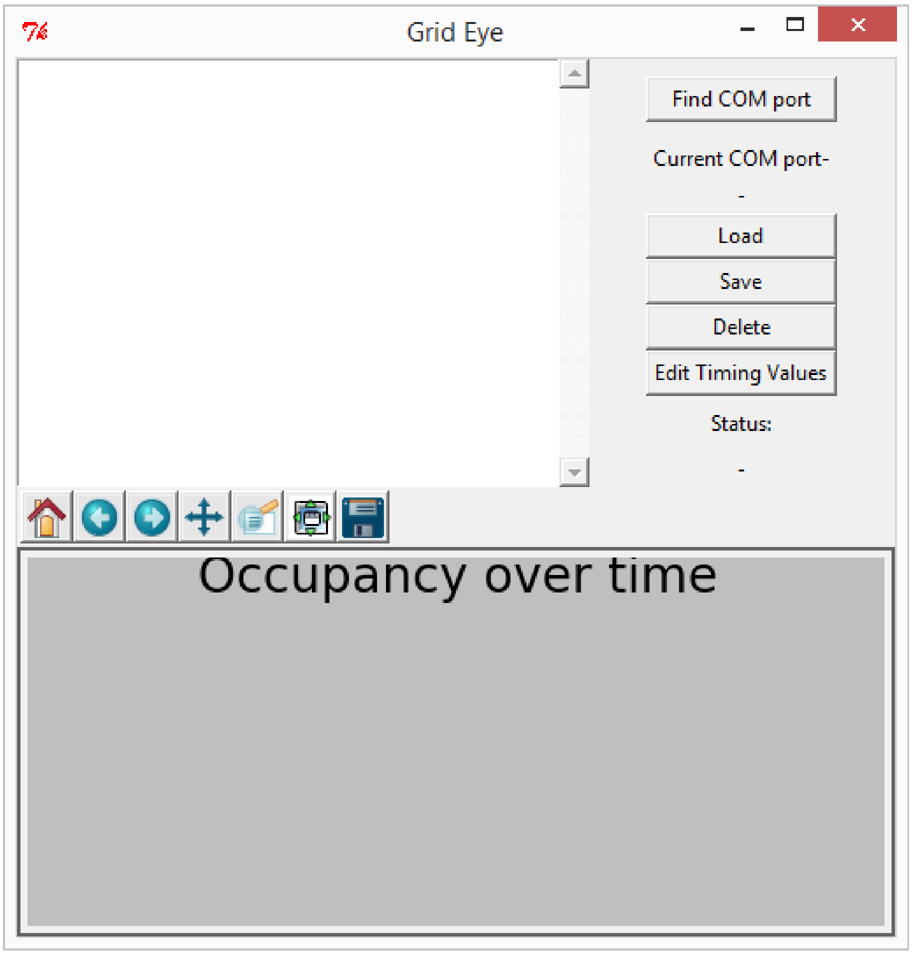


Figure 1: The software on startup

Edit Timing Values This allows you to change the Real Time Clock values in the data. Normally you would note the date and time that you started the device and then you can just write that into the pop up field (in the correct format). All times will be updated as if they started from that time.

Delete Use this option to delete a file from the SD card, it also has a 'Delete All' button in the pop up window. Deleting takes a bit of time and the only feedback is in the status window so be patient with it.

Problems Almost all problems can be solved by unplugging the device and closing the software, then trying again. This software is designed to be helpful but is by no means a finished product so there are bugs.

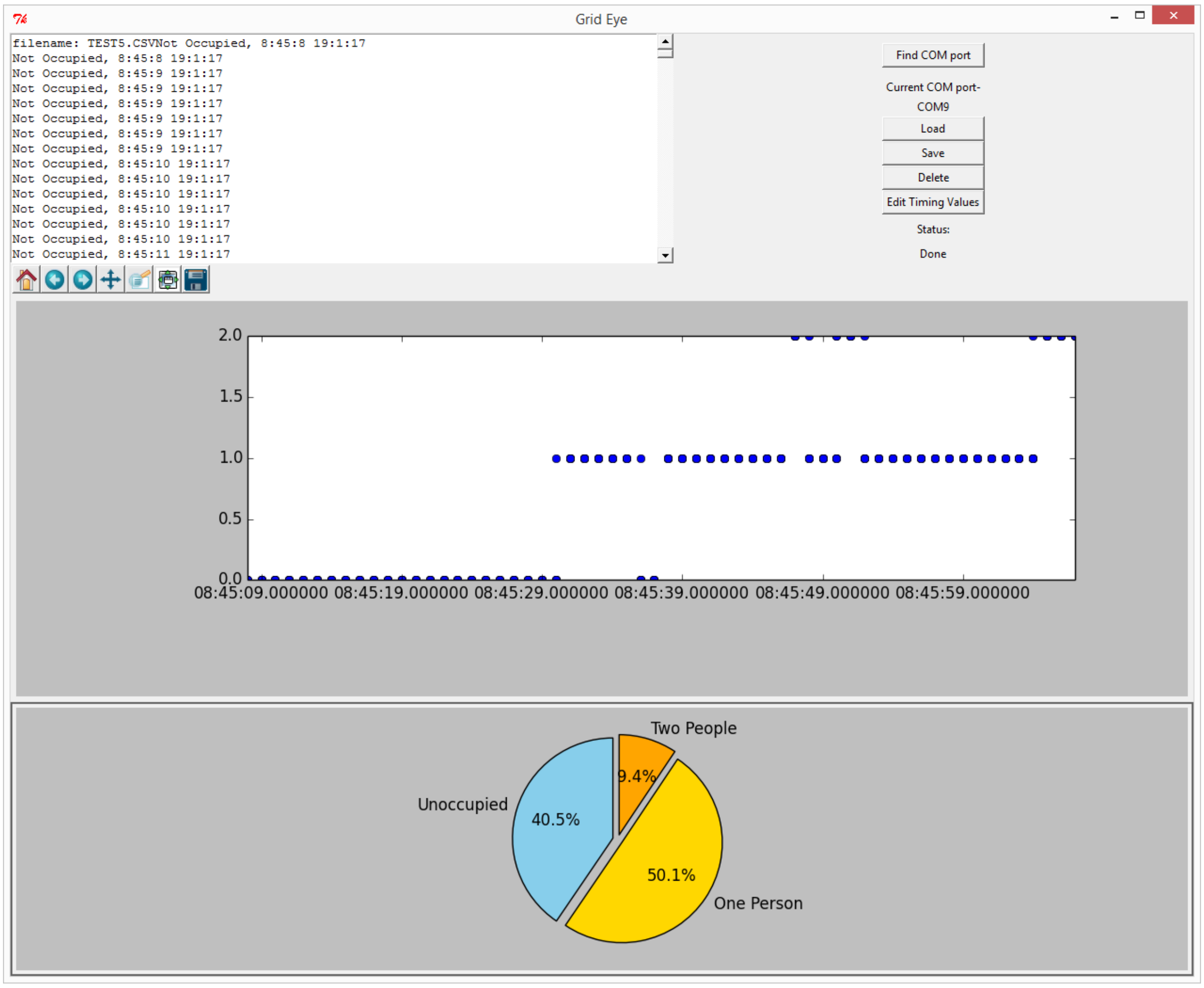


Figure 2: The software after loading

References

[1] arms22 Github User. Grideye, an arduino library for the grid-eye infrared array sensors. https://github.com/arms22/GridEye, 2015.

[2] Alex Beltran, Varick L. Erickson, and Alberto E. Cerpa. Thermosense: Occupancy thermal based sensing for hvac control. In *BuildSys'13 Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–8, 2013.

[3] G. Chabanis and D. Persegol. Smart imaging sensors for building & industrial applications. pages 11–13, 2013.

[4] John G. Cleary and Leonard E. Trigg. K*: An instance-based learner using an entropic distance measure. *Machine Learning Proceedings 1995: Proceedings of the Twelfth International Conference on Machine Learning*, pages 108–114, 1995.

[5] Robert H. Dodier, Gregor P. Henze, Dale K. Tiller, and Xin Guo. Building occupancy detection through sensor belief networks. *Science Direct Volume 38, Issue 9*, pages 1033–1043, 2006.

[6] Rikke Gade, Anders Jrgensen, and Thomas B. Moeslund. Long-term occupancy analysis using graph-based optimisation in thermal imagery. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3698–3705, 2013.

[7] Sunil Kumar Ghai, Lakshmi V Thanayankizil, Deva P. Seetharam, and Dipanjan Chakraborty. Occupancy detection in commercial buildings using opportunistic context sources. In *Work in Progress session at PerCom 2012, Lugano*, pages 469– 472, 2012.

[8] H. Burak Gunay, Anthony Fuller, William O'Brien, and Ian Beausoleil-Morrison. Detecting occupants' presence in office spaces: A case study. In *IBPSA eSIM 2016 conference*, 2016.

[9] Ebenezer Hailemariam, Rhys Goldstein, Ramtin Attar, and Azam Khan. Real-time occupancy detection using decision trees with multiple sensor types. *SimAUD '11 Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design*, pages 141–148, 2011.

[10] Philip C. D. Hobbs. A $10 thermal infrared imager. *Proc SPIE Vol. 4563*, pages 42–51, 2001.

[11] Tatsuya IZUMI, Hiroaki SAITO, Takeshi HAGIHARA, Kenichi HATANAKA, and Takanori SAWAI. Development of occupant detection system using far-infrared ray (fir) camera. Technical Report 69, Sumitomo Electric Industries, 2009.

[12] Suren Kumar, Tim K. Marks, and Michael Jones. Improving person tracking using an inexpensive thermal infrared sensor. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 217–224, 2014.

[13] Chris Lelievre. Github repository containing all the code done for this project. https://github.com/Chris-Lelievre/BRANZ-Occupancy-Sensor, 2017.

[14] Sensors online. Ulis and schneider electric unveil new advanced occupancy sensors. *Sensors online*, 2016.

[15] Thiago Teixeira, Gershon Dublon, and Andreas Savvides. A survey of human-sensing, methods for detecting presence, count, location, track, and identity. *ENALAB technical report*, 2010.

[16] Ash Tyndall. Towards a low-cost, non-invasive system for occupancy detection using a thermal detector array. Master's thesis, University of Western Australia, 2015.

[17] Joey Wilson and Neal Patwari. Radio tomographic imaging with wireless networks. *IEEE Transactions on Mobile Computing*, 9(5):621–632, 2010.

[18] Xandem. Xandem, monitor an entire house without cameras. Indiegogo crowdfunding project, 2015.

[19] Moustafa Youssef, Matthew Mah, and Ashok Agrawala. Challenges, device-free passive localization for wireless environments. In *Proceedings of the 13th Annual International Conference on Mobile Computing and Networking, MOBICOM*, pages 222–229, 2007.