Internet of Things for Room Occupancy Monitoring

Nuno Ferreira
nuno.p.ferreira@tecnico.ulisboa.pt
MEEC - Instituto Superior Técnico, Universidade de Lisboa, Portugal, December, 2019

***Abstract*— Determining the number and location of users inside a building is important information for both efficiency and security reasons. Many approaches can be used for this purpose including movement detection, identification of human bodies or changes in environmental conditions. In this work we used some of these approaches to try to find technologies that provide a good enough estimation of the occupancy of a room, meeting the criteria of being low-cost, non-intrusive and preserving the user's privacy. Therefore, we used vision-based technologies only for controlling the experiments and focused on other types of sensors for the proposed solutions. We started by a review some of the technologies that can be used to count the number of persons in a room, from which we selected four: Time of Flight(ToF), Thermal Camera, Millimetre-wave Radar(mmWave) and Air Quality, this last one measured by the concentration of Carbon Dioxide and Volatile Organic Compounds.**

As a preliminary experiment, we started by testing the ToF sensor and the mmWave Radar to count passing vehicles at a gate of a car park, for which specific hardware and software was developed. The next step was to address the problem of counting the number of persons inside a room. Specific hardware and software were developed for each selected technology that was tested and tuned in different experimental conditions. Finally, the four technologies were tested simultaneously in a real meeting room to compare their performances.

The best results were obtained with the Thermal Camera and the ToF based solutions. We also found a positive correlation between the decrease of Air Quality and the occupancy of a room, in the situation where no doors or windows were opened. The Thermal Sensor showed a greater accuracy than the ToF Sensor. However, the latter excels in cost-effectiveness, being particularly useful in scenarios where a low cost or low complexity solution is needed.

Keywords - Air Quality, People Counting, Occupancy Detection, Time of Flight, Thermal Camera, TI mmWave, IoT

I. INTRODUCTION

There is currently a growing need to make buildings as intelligent as possible. This requires the knowledge on how the building is used, so we need to know how many persons are inside the building and where they are. Measuring room occupancy is, thus, an important step either for security[1] or efficiency[2], since there are many applications that need a reliable and automated way to count people. One of the examples is the fact that with the increased concern about the air quality in enclosed spaces such as metro stations, classrooms, workplace locations, there has been a rising demand in studies that measure the air quality [3], [4], [5] (regarding quantities of pollutants) in various enclosed spaces, such as public transportation. However, in order to be able to properly determine and possibly predict the evolution of these values, we need to have reliable data, to be able to create a model for prediction.

There are many different methods for counting people that are being used such as mechanical methods like turnstiles. The major disadvantage of many of the methods is that they limit the flow of people in public areas. However, with the lower cost of technology, we can now use ideas such as Impulse-Radio Ultra-Wideband (IR-UWB) Sensors[6], infrared sensors (IR) [7] , thermal people counters [8], stereo [9], time of flight (ToF) [10], Wi-Fi[11], [12] or just simple Red Green Blue (RGB) Cameras [13].Video recording is highly discouraged nowadays due to security and privacy concerns, therefore we are left with mainly IR, ToF and thermal counters to answer the need of a good estimate of the number of people. Although there has been one new development lately available which is the Frequency Modulated Continuous Wave (FMCW) radar technology[14], [15], This work aims at answering the following research question: how good is the estimate of a low cost, non intrusive solution for occupancy detection, that can work together with indirect occupancy measures such as the air quality?

II. STATE OF THE ART REVIEW

A. *Air Quality Monitoring*

The air quality monitoring device that we will use in this work has the ability to measure the concentration of Volatile Organic Compounds (VOC) and Carbon Dioxide (CO2). This device was developed in a previous project to detect room occupancy. In this work we will try to see if VOCs and CO2 can also be used to count the number of persons in a room. VOCs are organic chemicals that have high vapor pressure at ordinary room temperature. The high vapor pressure is the result of a low boiling point, which causes large numbers of molecules to evaporate or sublimate from the liquid or solid form of the compound and enter the surrounding air. VOCs have multiple different sources, since they can origin from both human-made and naturally occurring chemical compounds. The second monitoring point is CO2. CO2 is widely known to be the most significant long-lived greenhouse gas in Earth's atmosphere. In this environment we will be observing, the fact that it is produced by people when they breathe, meaning there will probably be a link between higher concentrations of CO2 in a room and how occupied this room is. There have been some studies that have proved that it is possible to correlate occupancy with both these particle (VOCs and CO2) measures. [16], [17]

B. Vision based people counting

1) RGB Sensor: While vision based people counters are not ideal nowadays due to privacy concerns, they can benefit from the low-cost cameras and well tested image processing algorithms of today. Most of the implementations rely on background subtraction with a type of segmentation with k-means [18].K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. To adapt this algorithm to people counting, people are seen as concentrated shapes that can be extracted using k-means clustering, for example.

Background segmentation can be performed [18] on human detection systems by computing the pixel-by-pixel difference between the current frame and the background image, followed by a threshold.

The camera positioning is really important for people counting since it has a large impact in the result quality. For counting purposes, the zenithal positioning (which consists of a camera placed overhead) is the most adequate, since it reduces occlusion between objects, while also offering some advantages such as having a relatively constant size, eliminating the need for calibration. Another major advantage is that it addresses privacy concerns, since this positioning reduces face capturing. These types of methods manage to achieve accuracy of over 90% [18], [13] .

2) Depth Cameras: Depth capable cameras have been getting more accessible and are thus a viable choice for identifying people. This kind of technology is usually employed in conjunction with a regular RGB camera [13], because the added information of depth allows to overcome some of the shortcomings of just using a RGB camera. Some of these shortcomings are, as mentioned before, issues caused by changes in lighting, shadows and compound objects.

As seen in [13] they try to propose a method that is able to guarantee a high immunity to foreground detection errors and to overcome the problems mentioned before that mostly show up when there are a lot of people in the camera frame. Additionally [13], using shape recognition in embedded systems is computationally expensive, making it an unviable choice. Tracking is also recognized as an expensive feature. In this study, it is mentioned that using the depth information allows to detect more accurately the foreground than in comparison with just a standard RGB camera. The false positive rate is zero when depth information is used with a RGB camera, which improves the precision to 0.990 in combination.

3) Stereo Cameras: Using a stereo camera setup [9] that hangs in a zenithal position allows for a better segmentation of background and people. Unfortunately, the usage of dual cameras and its need for calibration means this setup isn't as sturdy and resistant to change than other proposed methods, given their added complexity.

C. WiFi People counters

Most sensors only work in fixed gateways or check-points, due to their limited area of effect. For example, a ToF sensor only works if the person crosses it and is based on a thermal approach that only counts people within its frame. This results in low accuracy [11].

The presented study [11] achieved a performance of 93% and exhibited robust performance against environmental changes with negligible power usage. This paper developed an in depth in the Rich Site Summary(RSS) studies on humans. This method uses the WiFi connection of smartphones as a counting tool.

Another study [12] using a Deep Neural Network(DNN) with 3 layers and a single WiFi transmitter and receiver, estimated up to 9 people in a room with 88% of accuracy, using signal correlation. This is not ideal overcrowded environments, since correlating both signals would be difficult.

D. Time of Flight

According to [10], ToF technology it is a non-contact measuring technique, that has been applied in a wide range of industrial appliances ranging from automatic assembly to quality assurance. These devices can provide depth information. They consist of a modulated light source such as a laser, a CMOS imager made out of an array of pixels and an optical focusing system. Two types of ToF depth sensors exist: the first class being represented by Pulse Modulation(PM) sensors. In this type of sensors the distance is computed directly from the time of flight using a high resolution timer that measures the delay between signal emission and reception. Depth measures are then obtained by:

$$d = \frac{TOF \cdot c}{2} \tag{1}$$

Where TOF is the time of flight and c is the speed of light. The second type of sensors are named Continuous Wave Modulation (CWM) sensors. The distance is calculated by the phase of the modulation envelope of the transmitted infrared light as received on a pixel. Using $s(t) = (\sin 2\pi * f_m * t)$ as model for transmitted light, where fm is the modulation frequency. The amount of light r(t) reflected by target is given by

$$r(t) = R\sin(2\pi f_m t - \theta) = R\sin(2\pi f_m(t - \frac{2d}{c})) \tag{2}$$

where θ is the phase shifting when the light returns to a sensor pixel, R is the amplitude of the reflected light and d is the distance between the sensor and the target. Therefore, d can be computed from θ as follows:

$$d = \frac{c\theta}{4\pi f_m} \tag{3}$$

PM sensors require a high resolution timer and a large bandwidth signal source to achieve high resolution measures, which means they are usually more expensive than CWM ones and better for long range applications. CWM devices are prone to aliasing problems, which can be diminished using multi frequency scanning. CWM new counters can get really high accuracy and they can provide an array of 64x64 distance measures.

ToF sensors have the advantage of being able to improve 3D Stereo vision people tracking [10] because they improve the cases where we have untextured scenes due to homogeneous objects or poor illumination. ToF sensors add some depth based on geometrical constraints and invariants. An approach based on ToF sensors requires no computation for 3-D scene re-construction and turns out to be independent of the degree of texture and the lighting conditions of the scene.

E. Thermal Imaging

Thermal imaging sensors [8] respond to emitted radiation, more so than reflected radiation. All objects emit heat by either conduction, convection and radiation. Radiation is the most important part for these kind of sensors, because objects continuously radiate heat with certain wavelengths, depending on the temperature of the radiating object and its spectral emissivity . As the object temperature increases, the radiation increases. The radiation emitted includes the infrared emission (which include wavelengths from 0.7 um to 100 um). These emissions are then detected by the thermal imager and made visible as an image in the form of a map of apparent temperatures.

Thermal imaging converts thermal radiation into a digital signal which is in turn converted into a visible image. To reduce cost we will use low resolution, 8x8 or 16x16, but via interpolation we can add higher resolution images for improved human perception. This interpolation does not add extra information to the image. Lower resolution images are also easier to be processed and can be done in an embedded system.

Results obtained with standalone thermal Imaging are approximately accurate to 5% according to [8]. While combining both methods (image and using a artificial neural network (ANN) for the IR imaging) got this percentage down to 3%. The sensor should be applied at entrance to count people in a narrow patch and then add those people as they move inside the room, in order to be able to count large groups of people. The big advantage of thermal imaging is the fact it can function well in low lighting situations, unlike purely vision based systems.

F. TI mmWave

First of all to understand the usage of the TI mmWave Sensors, we need to understand the underlying technology. The sensing technology used is named FMCW Radars, where the FMCW stands for Frequency Modulated Continuous Waves. These types of radars measure the range, velocity, and angle of arrival of objects in front of it.

A FMCW radar transmits a signal called a "chirp". A chirp is a sinusoid whose frequency increases linearly with time. [19]

:

$$f_{IF} = S2d/c \quad (4) \qquad d_{max} = Fs \cdot c/S \quad (5)$$

According to 4, an object at a distance d produces a certain. IF frequency. And, according to 5 , the Analog to Digital Converter (ADC) sampling rate Fs, limits the max range (d_{max}) to a certain value.

The Chirp Bandwidth and the IF Bandwidth are related. A sinusoid in time domain produces a peak in the frequency domain. In general the frequency domain is complex (i.e. each value is a phasor with amplitude and phase) which means we have the following: $A \cdot e^{(j\theta)}$

Both the transmitted and received signal, get input into a mixer and this which produces an IF signal with a constant frequency $\frac{S2 \cdot d}{c}$. A small motion in the object will change the phase of the IF signal but not frequency. The phase difference measured across two consecutive chirps can also be used to estimate the velocity of the object by:

$$v = \frac{\lambda \cdot \Delta\Phi}{4\pi \cdot Tc} \quad (6)$$

Where $\Delta\Phi$ is the phase shift between the two chirps and where Tc corresponds to the time that separates the two chirps. It is also possible to distinguish multiple objects that are equidistant from the radar, each travelling with different velocities using a Doppler FFT.

In what regards more in depth info of the TI mmWave device, ADC data corresponding to chirps are stored as rows of matrix. Range-FFT on each row resolves the objects in range. After this, Doppler FFT resolves the velocity of each object (i.e. objects at same distance but difference velocity can be distinguished). If they have same velocity but are different, you need more than one antenna to measure the angles of arrival.

Better knowledge of the technology will be needed if it is intended to try and change the clustering that TI already offers to its users, but since the IWR1642 device does most of the processing in the board before giving access to that data, it should not be needed.

1) Use cases: Currently Texas Instruments has several demos available using this technology, being the most relevant for the intended application the people counting demo[20] which allows for clustering and point cloud formation in a closed room with great success. It does this by checking what is static (and consists of the background) and what is moving and then, with a correct parameter setting, allows for people counting. Using this and modifying the supplied code to count people after they cross an imaginary line was the methodology used with this device

G. Choosing the devices to further develop

	RGB	Depth	RGB+D	Stereo	WiFi	ToF	Thermal	TI mmWave
Low-Cost	5	4	3	2	5	3	4	1
Performance	3	4	5	3	3	N/A(3)	4	5(theoretical)
Range	2	2	2	2	3	4	2	5
Complexity	3	3	4	5	3	2	3	4
Robustness	2	3	4	2	2	3	3	5
Availability	5	3	3	1	5	3	2	3
Total	20	19	21	15	21	18	18	**23**

TABLE I: Comparison of different methods

With the help of Table I, we can now decide which technologies have the most promise to develop. While RGB + Depth Camera have a great result, we still use the visible spectrum, which is something we are trying to avoid, plus the needed computer to process this data lead us to search for other solutions. WiFi Counter is good in theory, but with the rise of multiple devices belonging to the same person in one room makes for a difficult count. So we end with the Time of Flight Sensor, the Thermal Camera and the TI mmWave.

III. TIME OF FLIGHT

The time of flight alternative proved to be of interest mainly due to its simplicity of use, with the possibility of being robust. Since the first ToF sensor used only allowed to measure a single point, we decided that a simple car counting system would be a good way to test the possibilities of this type of sensors.

A. Car Counting

1) System Architecture and Algorithm: The Car counting system was installed on the entrance of the IST Campus in Alameda, counting cars that exited the campus. The system Architecture is composed by a TeraBee TR-EVO ToF sensor which is connected via I2C to a Raspberry Pi that is running a Python script for people counting. This Raspberry also has a camera that is activated when the Python script considers there is a car crossing, to serve as a GroundTruth. This Python script also interacts with a cloud service hosted in PythonAnywhere, where we display a webpage that contains the latest car crossing with a timestamp.

Algorithm 1 ToF Sensor Algorithm

```
1:  procedure PROCESS DISTANCE READINGS
2:      distance ← readi2c(trevo)
3:      if distance >= 0.5 then
4:          if distance < backgroundavg/2 then
5:              car = car + 1
6:              if (car > 2)&&(pic == 1) then
7:                  take_upload_picture()
8:                  pic ← 0
9:                  total_cars ← total_cars + 1
10:     else if distance < backgroundavg/1.3 then
11:         car ← 0
12:         pic ← 1
13:     if car == 0 then
14:         reads ← reads + 1
15:         total ← total + distance
16:         backgroundavg ← total/reads
```

For the Teraranger TR-EVO, a python library was created for easier interfacing with the sensor. The features of it are to read from the device and to change the i2c address of it. In terms of algorithm, as seen in Algorithm 1 it is a pretty simple algorithm with a certain number of flags that allow the Raspberry Pi to know whether it should count the reading to the background distance, or if the distance reading corresponds to the sensor being blocked. Regarding the algorithm, all the variables start with the value 0, and the portion of code shown corresponds to the while 1 loop that runs inside the program. The algorithm starts with the reading of the current value of the sensor. After this, we need to figure out if the sensor is reading an actual value (this means a value of 0.5m or more, since the minimum distance for this sensor is the 0.5m range). If that is the case, we then proceed to two different cases: if the distance read is less than half of the current background value (line 4), we increase the number of counts that the sensor has been blocked (we can't consider a pass with a single sensor block, because this would trigger a lot of false positives). In case this increase drives the variable to a count number of 3 and the flag to take a picture is active (line 6) then we register the count and upload the picture to the webserver, while making the picture flag zero, and increase the count of the total number of people counted. Line 10 of the algorithm serves to see if the person has crossed the sensor, making the value measured closer to its default background value. If that is the case, the count variable is set to 0 and the picture flag is back to 1 (meaning it can now take another pic when crossed).
Line 13 is the last statement, that means if the counter has been set (or its current value is 0), we update the current backgroundavg variable, via total and reads variable.
The application suggested is to use this sensor to count car entrances in one of the side doors of the university, since the sensor does well with big objects (that should trigger less false positives) and there is a road specifically for entering and another one for leaving, and the results shown in the next section are for cars leaving the University.

2) Results: We first obtained values of distance of the ToF sensor during an entire week. We then created a subset of data from one of the days and crossed the data obtained via the ToF sensor with the card readings of the exit gate in order to have a performance metric of this system. The day picked for this analysis was the 11th of July. We also have pictures of crossings, because sometimes the system denoted a car crossing, but the camera did not connect or take the picture.

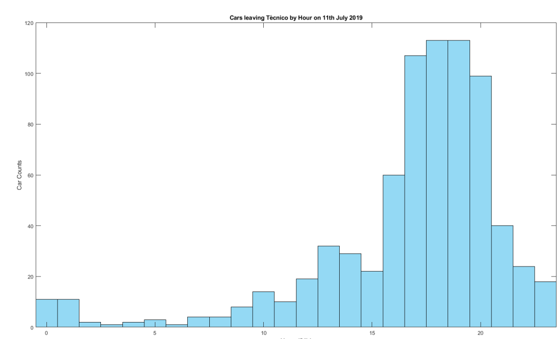

Fig. 1: Histogram of 11th July Data

Histogram 1 was plotted to show which hours of the day had more traffic and clearly they show that later on the day has the most activity regarding exiting the university, for a total of 747 cars during the day of 11th of July.

All this data needs to be compared with the system that controls the entrance of the campus. With the help of DSI (Tecnico's information technology services), the number registered by the system was 854 for the full day of the 11th of July. The data for the time of flight, had to be time synchronized with the system from DSI.

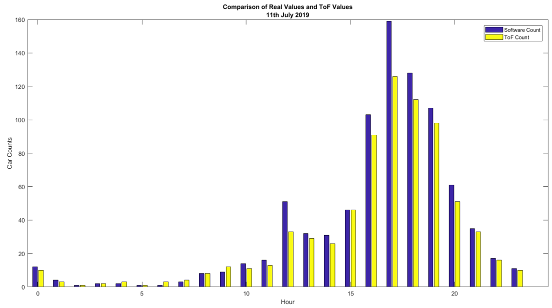

Fig. 2: Software data

As shown in Figure 2, the ToF follows mostly the Software data, but we also got some photograph confirmed cars that were registered but not caught by the software. The accuracy obtained by the ToF solution was 86.69%, which is an acceptable performance, but it would have been expected a better outcome. As it was expected most of the time what happens is that the ToF counts less occurrences than the Software, but there are some anomalies in low count cases, where the ToF counted more than the Software. This has raised some questions, so we turned into the validation pictures to assess whether this was a mistake or an actual good count. We deduced that the difference in bins is just a matter of seconds, which could be due to different clocks in both systems, making this, extra count from the camera, a non issue.

In conclusion, this system proved to have a satisfactory performance when there is the need to know the approximate flux of cars (with ideally two time of flight sensors in a single Raspberry Pi, in order to coordinate both entrances and exits, in case there are two separate paths for each).

B. People Counting

1) System Architecture and Algorithm: The architecture can be broken down into two hardware systems and a cloud server. The interaction with the VL53l1x sensor is made via I2C with an Arduino. This Arduino counts the entrances and exits with an algorithm and sends the output of the count via Serial Port to a Raspberry Pi Zero W, that publishes these values to a website, alongside storing the values in an SQLite Database. This Database is also stored server side for redundancy and to make possible the plotting of the results. The website displays the current count of people inside the room and the timestamp of latest crossing. The positioning of the system in the middle of the door is crucial, since the cone of view of the ToF sensor is very sensitive (bad positioning could make that people passing in the opposite corner of the door would not be detected).

For VL53L1X interface, we modified the provided libraries to provide adjustable ROI on the fly. Other than that, it is a pretty simple algorithm, inspired by the demo of STMicroelectronics. The counting algorithm example relies on a list of states that have to occur in a certain order to detect if a person has crossed the specified area and in which direction this area has been crossed. These states are stored in a list and compared to two default lists of states that represent how the area is crossed in two different directions. When no-one is seen in either of the two zones, the list of states is reset. If we consider that a person detected in the front zone equals 2, and a person detected in the back zone equals 1, the algorithm adds the value of the two states and stores the result as soon as it changes. Eventually, if the consecutive states in the list are 0, 1, 3, 2, 0 or 0, 2, 3, 1, 0 this means a person has been detected in one direction or the other, as described in the figures below. The algorithm validates a crossing event only when a person has fully crossed the two zones. It does not validate the event when the person remains for a long time under the FoV or when the person decides to return from the place he came from.

Algorithm 2 ToF People Counting Algorithm

```
procedure PEOPLE COUNTING TOF
    Set_ROI_Center(zone)
    distance ← vl53l1x_read_distance()
    if distance < Threshold_Distance then
        S_Det = 1
    if S_Det == 1 then
        detected = check_zone_detection()
        if detected == LEFT then
            if Current_zone_Status == S_Det then
                All_zone_Status += 1
            if R_zone_prev_Status == S_Det then
                All_zone_Status += 2
            L_zone_prev_Status == S_Det
        else if detected == RIGHT then
            if Current_zone_Status == S_Det then
                All_zone_Status += 2
            if L_zone_prev_Status == S_Det then
                All_zone_Status += 1
            R_zone_prev_Status == S_Det
        Event_Ocurred = 1
    if Event_Ocurred == 1 then
        Resize_Path_Filling()
        if R_zone_prev_Status == empty && L_zone_prev_Status == empty
then
            if Path_Track_Filling_Size == 4 then
                if Check_Entry() then
                    PeopleCount = PeopleCount + 1
                else if Check_exit then
                    PeopleCount = PeopleCount - 1
            else
                Update_Path_Track(All_zone_Status)
    Zone = Zone + 1
    Zone = Zone%2
```

Algorithm 2 is the one being used in the VL53L1X sensor. This is one iteration of the algorithm which reads the distance for one of the zones, since the algorithm consists in reading both zones determined by the ROI being set, in an alternate way. The first step is to set the ROI of the current zone being analysed(either the left or the right zone). With this ROI, now we get the distance reading from the current zone. This distance is then evaluated to see if it has crossed the desired threshold (the distance reading that we consider to be a person). If that is the case, the status flag Someone Detected is set to 1. After, in case this flag is active, we need to check whether this was a hit on the left or the right zone. If the zone that was detected was the left zone, and the current zone status is that someone was detected in that zone, the Status for all the Zones is incremented to 1. This is a control variable that can have values ranging from 0 to 3. When 0, it means that no one is being detected in all the zones. When 1, it means that there is someone detected on the left zone. When 2, it means detection on the right zone. For 3, it means there is detection in both zones at the same time. Going back to line 2, if the previous status of the right zone was of detection of a person, then we add 2 to the overall status counter. The distinction between Right and Left is needed due to the algorithm running the same for both zones, but it runs a certain part if the zone being analysed is a certain one. That's why the same line of thought is done for the line 2 and below. If someone was detected, this algorithm also puts the Event Occurred flag to 1, which then triggers a line of events. In line 2, we resize the path filling size variable, which only happens if its size is smaller than 4, since only when it reaches 4, the person passage is analysed. The following line, shows that in case both right and left zones previous statuses are considered empty, the algorithm checks whether there was an entrance or leaving of the room, depending on the Path Track format. The path track is the history of the sensor activation. For example, a path track of 0 1 3 2 counts as an entrance and increase in people counting, since the history of the sensor was an activation of the left zone, activation in both zones, and then finally activation only in the right zone. But, as mentioned in line 2, the path track is only updated if we do not reach the path track filling size of 4. The algorithm ends with switching zones to do the same, all over again for the next zone.

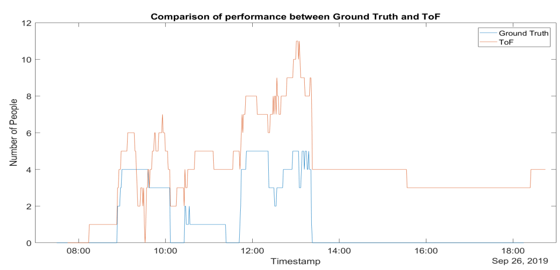

Fig. 3: ToF People Count comparing with Actual Count

2) Results: Using the ToF solution proposed revealed to be an adequate solution. The comparison isn't being made against the ground truth values since those values are only for people in the meeting table, but the macro analysis, seen in Figure 3 (as in when a lot of people enter or leave) match up. Some issues were expected, since this is a single ToF sensor measuring two zones, so if two people cross the zones with a small space between them, they can be incorrectly identified. A faster processor than the one on the Arduino Uno, could have also helped.

The results shown in Figure 4 clearly note a that more entrances were counted than exits, which is not correct (that would mean that in the end of the day, there would be people inside the room). This denotes that either the sensor has a better performance in one of the directions, or that people have different behaviours when leaving and entering the room. This last hypothesis seems a possibility, since that in most conference rooms, people tend to enter the room

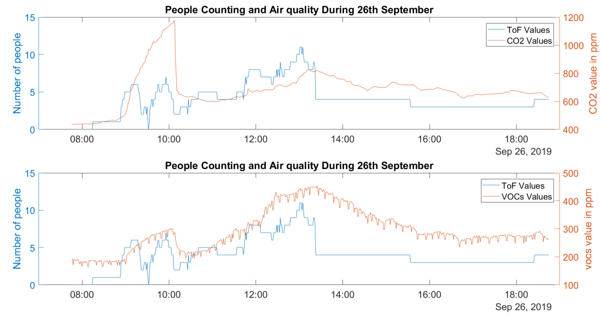

Fig. 4: Air Quality results compared to ToF People Count

slowly and with space in between them, while they tend to leave in a more rushed manner, with less gaps in between persons.

The results are pretty satisfactory since it is possible to notice a relationship with the increasing values of people count (when it starts increasing around 9:00, both CO2 and VOCs values start increasing), having a drop off a little after 10:00, when the counting system determined that people left the room. A little after this, we started seeing another increase in VOC and CO2 around 12:00. The clearly noticeable part of people leaving the room, can be seen around 13:10, making the CO2 and VOCs value peak and slowly get lower over the next few hours. The person being recorded leaving at around 16:00, probably entered and left the room very fast, making this +1/-1 count happen, which is one of the problems of the ToF sensor. In a macro point of view, it is possible to see that the occupation is clearly shown by the air quality and that the clear increases in ToF both in entering and leaving the room predate changes in the air quality.

C. THERMAL CAMERA

Two different approaches were considered: a static one to be implemented in conference rooms that counts the current number of people in a table; a people flow counter that is placed at the entrance of a door that counts how many people enter and leave the room in which it is placed. This allows to have a number of people to cross with air quality monitoring, in order to determine the occupancy of a room. We are going to focus on the conference room system, since it's the one that was further developed.

1) System Architecture and Algorithm: The System is composed by a thermal camera is connected to the Raspberry Pi via I2c, with a C++ library with some modifications made in order to save the values of temperature to files, enabling the Python script to be able to process them. The output of the library by itself is only an animated PNG(Portable Networks Graphic) (APNG) file, which is basically an image sequence. Then the Python script processes the temperature values as seen in 2 and posts the count result to a cloud server that updates the webpage for the user to see. The webpage has a picture of the current thermal image, alongside a graph of the last days of people counted.

Algorithm 3 Thermal Camera Algorithm

```
 1: procedure PROCESS IMAGE
 2:     Size ← 3
 3:     numpic ← 8
 4:     temperatures ← read(file)
 5:     for i = 0; i < numpic do
 6:         timage[i] ← temperatures[i]
 7:     threshold ← median(timage)
 8:     for i = 0; i < numpic do
 9:         for x = 0; i < columns do
10:             for y = 0; i < lines do
11:                 if timage[i][x][y] < threshold then
12:                     timage[i][x][y] ← 0
13:         result[i] ← connectedcomponent(timage[i])
14:         if result[i].size < Size then
15:             (remove(result))
16:         numppl[i] ← size(results[i])
17:     pplcount ← mode(numppl)
```

The Algorithm 3 for processing the thermal image is as follows: from a 32 frames per second video, we pick out the first 8 (therefore the numpic variable in the algorithm), in order to reduce the timeframe of analysis (and to reduce change in the pictures being analyzed). Then, we need to find what is considered background, in order to apply a threshold to the picture. This is made by median analysis of all pictures together (line 7 of the algorithm). Then, we apply the obtained threshold to each individual frame in order to have a preliminary analysis. This should leave us with only the objects of interest, which are the ones with the highest temperature. But sometimes there are some sensor faults, therefore there are small spots with high values that have high temperature. So after this it is run a connected components analysis in order to find objects in the picture. But not all objects in the picture are people, since there are some objects in conference rooms, for example, that also emit heat, such as laptops or projectors. Fortunately, these objects have a smaller footprint than people, so for the connected components we remove those that have a pixel size below a threshold . The number of connected components left give us the number of people in the room. After this, the mode of values of people in the 8 frames is classified as the number of people inside that room in that timeframe. Afterwards we send this number server side via POST to a server in order to display the information on a website, which displays the current image of the room, plus a graph of the count of people (last 3000 points, due to limitation of the highcharts library used). The server also updates the sqlite database (not sure if this database is server side or client side).

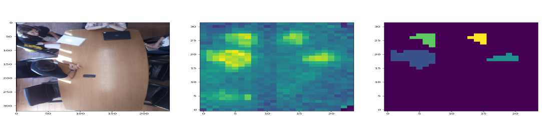

Fig. 5: Visible Image and corresponding Thermal Image

2) Results: Once again, this location was chosen to test the performance of the thermal camera in the same way as before. The new and improved version of the algorithm with adaptive thresholding based on temperature was used in order to increase performance but while reviewing the images there was something odd we noticed. In Figure 5 we can see the various elements of the Thermal Sensor. On the left, we have the visible picture that is the basis for performance analysis of the thermal sensor. The middle picture, is the original thermal picture, without any type of thresholding or adjustment. And on the right it is possible to see the outcome of the algorithm, with four identifiable blobs corresponding to each one of the persons in the picture. To note, that the thermal camera image is slightly tilted compared to the visible image and due to the wider FoV of the thermal camera, it captures some zones that the visible camera for ground truth does not capture.

There is also a spot on the floor that receives the sun gets too hot, which leads to false detection on the algorithm. So we had to change the algorithm to remove counts in that general area.

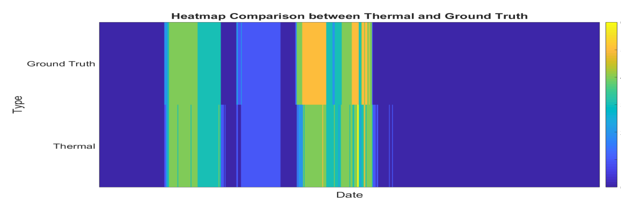

Fig. 6: Heatmap Comparison

As for this sensor, in Figure 6 we have an accuracy of 86.07% which is pretty good since during the lunch hour the room is densely populated. Even with this kind of accuracy, the thermal footprint of the image lets us know if the room is very populated or sparsely populated. After this we need to see if the results compare in a positive way with the air quality measurements. This accuracy data is more clearly represented in Table II.

Number of errors(+/-)	Percentage
0	86.07%
1	11.30%
2	2.32%
3	0.31%

TABLE II: Performance of Thermal Camera

With the known thermal performance results, we can safely say that it can be used to asses the occupancy of a room. So it is clearly interesting to cross the value of the thermal results with the air quality data, to see if there is some kind of correlation.

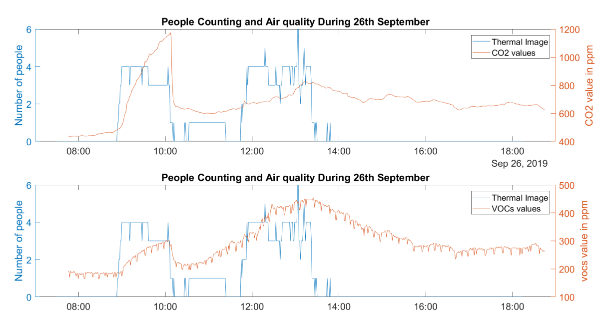

Fig. 7: Thermal Sensor compared with air quality

Once again, in Figure 7 we can notice a clear correlation between the air quality and the occupation obtained via thermal images. Both CO2 and VOC values have an increase when the room start being occupied, and start decreasing after it becomes vacant. Once again this method proved to be a reliable indicator, matching the air quality.

IV. TI MMWAVE

The TI mmWave was the technology that showed the most promise when doing some preliminary testing, as well due to its use in multiple papers such as [21], [22] . So it was decided to test it out for not only people counting in indoors environments but also for car counting purposes, in order to have a comparison system for our ToF to be evaluated against. Also since the setup architecture is pretty simple in both cases (composed mainly of a computer where the mmwave is connected with a database), it does not need a more thorough explanation.

A. Car Counting

Using the TI mmWave as a counter for passing cars in both lanes is possible by using the prepared sample code by Texas Instruments, which in theory will let us compare the performance of this sensor with the Time of Flight one.

The TI mmWave was mounted outside in such a position that allowed the radar to record both lanes of traffic that occur in the entrance of Instituto Superior Técnico - Both entering and exiting the university. And while it was recorded in the data the number of cars entering, only the exiting number of cars will be used for comparison purposes with the ToF Sensor.

1) Results: In order for data to be comparable to the work developed in section III, the data chosen for analysis was for the 11th of July. It is only possible to gather the data from a certain timeframe and analyse it in terms of total cars, since the software did not provide an exact timestamp. When replaying back data, the software counted 487 cars in this timeframe and with hand-made annotations (noticing when the centroids would form, but would not count as crossing the exit), the count is 648.

The ToF sensor in the same day (11th of July) counted 746 vehicles, which corresponds to a precision os 87.35%. This is a better result than the one obtained by the mmWave prototype. The results aren't directly comparable, because while the ToF sensor is doing the detection when the vehicle is stopped, and with a low sample rate (the vehicle is only stopped a few seconds while the driver passes his card), the mmWave prototype has a job that is several levels of complexity above the task being done by the ToF. In the detection scene of the mmWave prototype, both the sensor and the DSP acquire the targets while moving, and track them continuously in a vision field that has several dozens of square meters.

The results of the mmWave in a first analysis are well below the expectations with an error of 206 counts(648 detections in 854 vehicles que left the zone), although having in consideration that we have a portion of the park (112 parking spots) that we are not counting, since the vehicles leave this parking zone to the exit, not crossing the main road being monitored. This park also has typically a rotation rate that is superior to 1. In an hypothetical scenery that the park was only occupied and unoccupied the 112 parking spots, the mmWave detection rate would be of 87.3%, which is a similar value to the ToF sensor that does not suffer from this uncertainty.

The prototype that we built allowed to obtain in a reliable and regular way data throughout a long period of time, which was only limited because the installation caused some insecurity to the security guards, due to the computer running the system being a potential thing for people to steal. Also the necessity of using a powerful computer for the processing, constant electrical power and the system complexity(which took longer than initially foreseen to be stable), have made the use of the mmWave for car counting in other scenarios not worth it.

Having these thoughts in mind, the results are probably due to setup characteristics and chirp selection, since the TI mmWave (and FMCW radars in general) are really dependant of the environment where you setup them, besides having a correct chirp configuration for the current use. The main problem we could find was that sometimes cars could not be correctly identifying if they were too close, or sometimes there was a large pointcloud that was not classified as a car. Real world performance is always different to test setups, and probably the position of the radar, where it was lower than the cars it was detecting, did not contribute for the cases where two cars were closely following each other (the hypothesis is that most cars that were not counted happened during the most dense hours, in terms of car crossings). The main advantage with this system is that the radar can detect both incoming and outcoming lanes.

B. People Counting

For counting people using the TI mmWave there were a lot of promising ideas that could be used. Firstly, we thought of using the PeopleCountingDemo just as it comes out of the box, but we noticed this would not be enough, even when using the people in box counting option. This is because, in the software, people could stop being detected, if they did not move enough, since the radar only detects moving objects. This would make for an incorrect count. So the option chosen for tracking people in and out of the room was to check when the centroids being detected crossed a certain crossing zone, and the direction of crossing determined if they were entering or exiting the room.

The counting algorithm fares pretty well to count the people exits, but since the setup in the room makes that the centroid is only tracked very near the moment the person enters the room, sometimes it is only generated after the person has already crossed the line that counts the crossings.

1) Results: The people counting demo has several configurable parameters, most notably the location of the line to be crossed. This is one of the most important parameters for good performance, because it must be placed properly to detect people as they cross the door entrance.

Unfortunately, for the mmWave, the results weren't as good as we were expecting, but we clearly counter the exits from the room more successfully. This is due to the problem mentioned before, that the positioning of the mmWave sensor was not ideal.

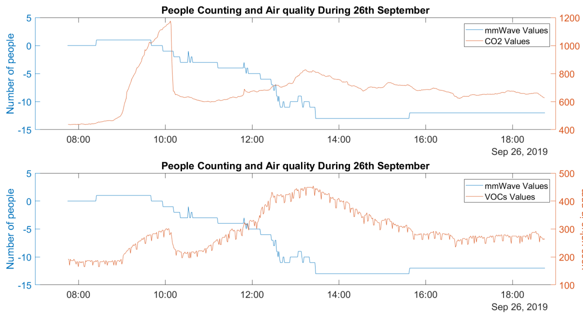

Fig. 8: TI mmWave compared with air quality

In Figure 8, we can clearly see that the people count keeps moving towards negative numbers, this is due to the fact that this sensor is detecting mostly the people that leave the room. It was expected not an ideal performance from the radar, since they work best in open air environments, mostly due to the amount of reflections that happen in closed environments. Even then it is possible to notice the decreases in CO2 and VOC with the decrease in the counts on the mmWave, noticeable around the 10:00 period and the 13:00 period.

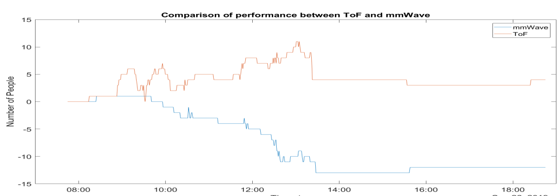

Fig. 9: TI mmWave compared ToF Sensor

We figured that a comparison between counts in both ToF sensor and mmWave would be interesting since they both compared entrances and exists. We see that while spare entrances work okay, the ToF is an overall better choice with the capability of counting entrances in a more consistent manner. The leaving of the room can be seen in both cases, which means that the exits that were counted with the mmWave were also counted with the ToF.

V. AIR QUALITY

We started by measuring the air quality inside the room and seeing how it compares to the number of people inside it. Then, if it is possible to validate that the air quality can be positively correlated to the number of people inside the room, we can use the air quality graph to compare how well our devices performed. The first analysis is the plot of both VOC (Volatile Organic Compounds) and CO2 (Carbon Dioxide) in ppm (parts per million) during a 24 hour period.

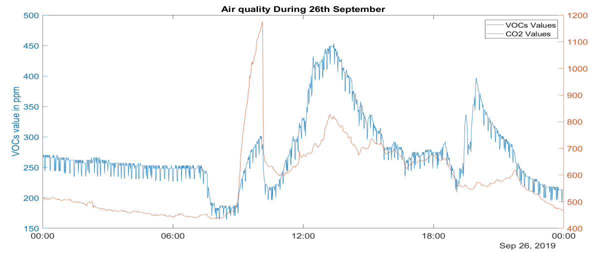

Fig. 10: Air Quality during 26th September

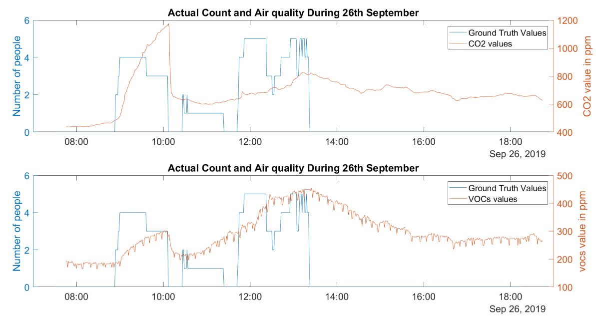

Fig. 11: Ground Truth Comparision with Air Measurements

In Figure 10 we can notice that both CO2 and VOC stay relatively constant until around 07:00 but while the VOCs drop, the CO2 stay constant. This can be explained by opening the doors of the room but not going inside, since VOC can be dust being released after opening of the doors. But when they both start to rise is when the room starts being occupied (proved by the people occupation graph in Figure 11). It is also possible to see that while the first meeting has a much higher CO2 concentration, but the VOC did not increase much, at lunch hours, when people eat and release particles of food to the atmosphere, the VOC increase much more than during the first meeting. But people were using the room, as seen once again in Figure 11. It could also be that during lunch the door was left open so the CO2 concentration was not as high, or perhaps the smaller duration of this event did not allow the CO2 to accumulate as much. At last, notice the decrease in CO2 in an approximately constant manner during the rest of the day, but a peak in the VOCs could be explained by someone releasing dust particles in the room (around 20:00). The CO2 starts decreasing in a bigger peak around 22:00, which makes sense since the room being monitored is at ground level. Since CO2 is an heavy gas, it accumulates at the bottom, so while the room might not be occupied, if there are people inside the building, the CO2 level only starts rapidly decreasing after they all left.

This small comparison between air quality monitoring and Ground Truth of the room occupancy shows that it is possible to determine if a room is being occupied with some uncertainty added to the mix. For example, just looking at the CO2 graphs, one would probably conclude that the first meeting had higher occupancy than the one that occurred during lunch. But using our people count solutions, one could see that is not the case, which is also confirmed by the ground truth. So ideally, the air quality measurements support another counting system.

VI. CONCLUSIONS

	ToF	Thermal	TI mmWave
Low-Cost	(3)5	(4)3	(1)1
Performance	(3)3	(4)5	(5)2
Range	(4)3	(2)2	(5)5
Low Complexity	(4)4	(3)3	(2)1
Robustness	(3)3	(3)3	(5)3
Availability	(3)4	(2)3	(3)2
Total	(18)**22**	(18)19	(23)14

TABLE III: Comparison of different methods

Table III is the updated version of the comparison table I, reduced to the implemented technologies. We can see the results for the performance and value proposition of each technology. The values are rated from 1(worst) to 5(best) and the values in between parenthesis are the values from the theoretical point of view. The worst performer in comparison to theory was the TI mmWave. We expect this to happen mostly due to the added complexity of this device, that is most probably better used when building from the base and not just trying to adapt already available demos. Both the Thermal Camera proposal and ToF had their strengths. For car counting purposes, the ToF presented a simple solution achieving over 86% accuracy, as seen in chapter III. For indoor counting, the ToF could be correlated positively with the air quality. We noted that it could correctly monitor people's entrances in sparse situations, despite having more difficulties when multiple people were leaving the room at the same time. Even then, it retained its ease of use and low cost. With most of these solutions, only a Raspberry Pi Zero W is required to post the data to the web (due to the low processing power needed) and a microcontroller capable of handling the ToF data. The Raspberry Pi Zero W can even be omitted if using a microcontroller with some IoT capabilities, such a nb-IOT support or SigFox, for example. The Thermal Camera solution is the best choice if the exact people count is desired, since the use of adaptive thresholding via room temperature yielded some good results. Once again, we obtained over 86% of people accurately counted This number can be further elevated if more data sets are used to fine tune the thresholds to the room temperature where the sensor is being installed. Being a thermal sensor, it is limited by the environmental heat sources that are hard to avoid, such as sunlight reflection or high laptop temperatures, creating a large hotspot. Finally, we can address our research question. A low cost occupancy detection solution can be built, with an accuracy up to 86%.

REFERENCES

[1] Duc Fehr, Ravishankar Sivalingam, Vassilios Morellas, Nikolaos Papanikolopoulos, Osama Lotfallah, and Youngchoon Park. Counting people in groups. *6th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009*, pages 152–157, 2009.

[2] Kemal Akkaya, Ismail Guvenc, Ramazan Aygun, Nezih Pala, and Abdullah Kadri. IoT-based occupancy monitoring techniques for energy-efficient smart buildings. *2015 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2015*, pages 58–63, 2015.

[3] Khaled F Salama, Rashed F Alhajri, and Abdulrahman A Al-anazi. Assessment of air quality in bus terminal stations in Eastern Province , Kingdom of Saudi Arabia. 4(5):1413–1418, 2017.

[4] Simone Meinardi, Paul Nissenson, Barbara Barletta, Donald Dabdub, F. Sherwood Rowland, and Donald R. Blake. Influence of the public transportation system on the air quality of a major urban center. A case study: Milan, Italy. *Atmospheric Environment*, 42(34):7915–7923, 2008.

[5] Mili Weng and Xin Jin. Study on the air pollution in typical transportation microenvironment: Characteristics and health risks. *Journal of the Air and Waste Management Association*, 65(1):59–63, 2015.

[6] Jeong Woo Choi, Sung Ho Cho, Young Soo Kim, No Joong Kim, Soon Sung Kwon, and Jae Suk Shim. A counting sensor for inbound and outbound people using IR-UWB radar sensors. *SAS 2016 - Sensors Applications Symposium, Proceedings*, pages 528–532, 2016.

[7] K. Hashimoto, K. Morinaka, N. Yoshiike, C. Kawaguchi, and S. Matsueda. People count system using multi-sensing application. *International Solid State Sensors and Actuators Conference*, 2:1291–1294, 1997.

[8] I. J. Amin, A. J. Taylor, F. Junejo, A. Al-Habaibeh, and R. M. Parkin. Automated people-counting by using low-resolution infrared and visual cameras. *Measurement: Journal of the International Measurement Confederation*, 41(6):589–599, 2008.

[9] K. Terada, D. Yoshida, S. Oe, and J. Yamaguchi. A counting method of the number of passing people using a stereo camera. *IECON'99. Conference Proceedings. 25th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.99CH37029)*, 3:1318–1323.

[10] Alessandro Bevilacqua, Luigi Di Stefano, and Pietro Azzari. People tracking using a Time-of-Flight depth sensor. *Proceedings - IEEE International Conference on Video and Signal Based Surveillance 2006, AVSS 2006*, 2006.

[11] Haochao Li, Eddie C.L. Chan, Xiaonan Guo, Jiang Xiao, Kaishun Wu, and Lionel M. Ni. Wi-Counter: Smartphone-Based People Counter Using Crowdsourced Wi-Fi Signal Data. *IEEE Transactions on Human-Machine Systems*, 45(4):442–452, 2015.

[12] Yen-kai Cheng and Ronald Y Chang. Device-Free Indoor People Counting Using Wi-Fi Channel State Information for Internet of Things. 2017.

[13] Luca Del Pizzo, Pasquale Foggia, Antonio Greco, Gennaro Percannella, and Mario Vento. Counting people by RGB or depth overhead cameras. *Pattern Recognition Letters*, 81:41–50, 2016.

[14] A. Wojtkiewicz and R. Rytel-Andrianik. Optimal detection and estimation in FMCW radar. *14th International Conference on Microwaves, Radar and Wireless Communications, MIKON 2002*, 3:778–781, 2002.

[15] Lapland U A S Publications. *State-of-the-Art Review on Automotive Radars and Passive Radar Reflectors Arctic Challenge research project*. 2018.

[16] Khee Poh Lam, Michael Höynck, Bing Dong, Burton Andrews, Yun Shang Chiou, Rui Zhang, Diego Benitez, and Joonho Choi. Occupancy detection through an extensive environmental sensor network in an open-plan office building. *IBPSA 2009 - International Building Performance Simulation Association 2009*, pages 1452–1459, 2009.

[17] Tobore Ekwevugbe, Neil Brown, and Denis Fan. Proceedings of the 2012 6th IEEE International Conference on Digital Ecosystems and Technologies: Complex Environment Engineering, DEST 2012. *IEEE International Conference on Digital Ecosystems and Technologies*, 2012.

[18] Suellen Silva De Almeida, Victor Hugo, and Cunha De Melo. An Evaluation of two People Counting Systems with Zenithal Camera.

[19] Cesar Iovescu, Radar Applications Manager, and Sandeep Rao. The fundamentals of millimeter wave sensors. 2017.

[20] Ti mmwave people counting demo. https://training.ti.com/people-counting-demonstration-using-ti-mmwave-sensors.

[21] Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. mID: Tracking and Identifying People with Millimeter Wave Radar. pages 33–40, 2019.

[22] orđe P. Glavonjić, Ivan M. Milosavljević, Dušan P. Krčum, Veljko R. Mihajlovic, and Lazar V. Saranovac. Person detection counter based on mm-Wave radar technology. *IcETRAN*, pages 4–7, 2017.