Heat-sensitive paint for thermal camera calibration

**Project ID:** MWH02b-22

**Supervisor:** Mow Wai Ho

**Author (Student ID):** LI Hoi Him (20585898), LI Hong Yin (20496853), LO Yat Hei (20692574),

**Date:** 13 September, 2023

**Main Objective**

The project aims to design and develop an alternative method to calibrate thermal cameras at a longer distance and at a lower cost with the use of economical heat sensitive paint. With its property of changing colour according to the temperature of the object it has been applied to, a large surface of paint can tell the thermal camera at a distance the temperature of the object. By combining temperature information from the paint, and the temperature received directly from the object, the camera can be calibrated accurately at a longer distance. As we can tell the temperature of the heat source by the change of the heat sensitive paint colour, we can conveniently calibrate the thermal camera based by calculating the difference between the activation temperature of the pain and the detected temperature of the calibration area.

#### Objective Statements

* 1. To set up a calibration area with the use of heat-sensitive paint
  2. To calibrate thermal camera by tracking colour of heat sensitive paint
  3. To track and display temperature of people at different distances

Table of Contents

[Abstract iv](#_Toc132804254)

[SECTION 1 – Introduction 1](#_Toc132804255)

[1.1.  Background and Engineering Problem 1](#_Toc132804256)

[1.2.  Objective 2](#_Toc132804257)

[Objective Statements 2](#_Toc132804258)

[1.3.   Literature Review of Existing Solutions 2](#_Toc132804259)

[Multi-Point Calibration 2](#_Toc132804260)

[Kiosk 3](#_Toc132804261)

[Temperature Sensor and Camera Feedback 3](#_Toc132804262)

[SECTION 2 – METHODOLOGY 4](#_Toc132804263)

[2.1 Overview 4](#_Toc132804264)

[2.1.1 System Description 4](#_Toc132804265)

[2.1.2 System Block Diagram 5](#_Toc132804266)

[2.1.3 Component List 5](#_Toc132804267)

[2.1.4 ECE Knowledge 6](#_Toc132804268)

[2.2 Objective Statement Execution 7](#_Toc132804269)

[2.2.1 Setting up the calibration area with Heat sensitive paint 7](#_Toc132804270)

[2.2.2 Calibrating thermal camera by tracking colour of heat sensitive paint 10](#_Toc132804271)

[2.2.3 Track and display temperature of people at different distances 15](#_Toc132804272)

[2.2.4 Displaying final results 17](#_Toc132804273)

[2.2.5 Code 18](#_Toc132804274)

[2.3 Main Objective Evaluation and Discussion 30](#_Toc132804275)

[MLX90641 31](#_Toc132804276)

[AR320 31](#_Toc132804277)

[Watson’s dual-mode infrared thermometer 32](#_Toc132804278)

[Results 32](#_Toc132804279)

[SECTION 3 – CONCLUSION 34](#_Toc132804280)

[Appendix A – Final Project Schedule 36](#_Toc132804281)

[Appendix B – Budget 39](#_Toc132804282)

[Appendix C – Meeting Minutes 40](#_Toc132804283)

[Appendix D– Group Members’ Contribution 52](#_Toc132804284)

[Appendix E – Deviations from the proposal and progress reports and supporting reasons 55](#_Toc132804285)

**List of Illustrations**

[Figure (1): Thermal camera calibration by heat sensitive paint and temperature measurements. 5](#_Toc132803709)

[Figure (2): Result of 3 HSP turning into brighter colours under extreme 7](#_Toc132803710)

[Figure (3): Setup of our HSP panel 10](#_Toc132803711)

[Figure (4): Image output of the thermal camera 11](#_Toc132803712)

[Figure (5): Logitech C310HD Webcam 11](#_Toc132803713)

[Figure (6): Placement of thermal camera right next to the webcam 12](#_Toc132803714)

[Figure (7) & (8): Thermal Image and RGB image of a person standing at 3.5m away. 13](#_Toc132803715)

[Figure (9): Example of program indicating the tracking area 14](#_Toc132803716)

[Figure (10): Cropped image of said designated calibration area 14](#_Toc132803717)

[Figure (11): Image of a MLX90641 31](#_Toc132803718)

[Figure (13): Image of Watson’s dual-mode infrared thermometer 32](#_Toc132803719)

[Figure (14): Comparison between various thermometers at different ranges 33](#_Toc132803720)

**List of Tables**

[Table (1): List of components 5](#_Toc132810268)

[Table (2): Comparison between one-use warm packs and electrical warmers 9](#_Toc132810269)

Table (3): Comparison between various thermometers at different ranges ………………………………………………………….………. 32

[Table (4): Final Project Schedule 38](#_Toc132810270)

[Table (5): Budget 39](#_Toc132810271)

# Abstract

To create a thermal camera thermometer system affordable by the average consumer to improve public health safety, we developed a low-cost calibration system for thermal cameras such that they could accurately read body temperatures of different people in any environment by utilizing heat sensitive paint, which changes colour at a certain temperature. We applied OpenCV libraries using Python programming language and used Raspberry Pi to run our low-end thermal camera. We also utilized image processing to read our calibrator, resulting in a calibration system that can accurately measure each person’s body temperature in sight with minimum cost and set up complexity. Users only have to stand next to the calibrator in view of the thermal camera to have their body temperature read automatically by the camera system.

# SECTION 1 – Introduction

## 1.1.  Background and Engineering Problem

Thermal cameras are imaging devices that measure the intensities of infrared radiation in view, unlike regular cameras that produces images based on the intensities of visible light. While every object in the universe emits radiation, the amount of radiation an object emits varies with its temperature.  Given the range in temperature on Earth, most objects emit specifically infra-red radiation, which are invisible to the naked human eye.

Since thermal cameras are able to pick up infra-red radiation to capture images even in the absence of visible light, they have been widely used for night vision-related equipment. Meanwhile, thermal cameras are also capable of detecting irregular hotspots within objects.  Defective electronics in a complex circuit would generate excessive amounts of heat, while some heat sensitive parts would malfunction and be permanently damaged if it becomes too hot.  Thermal cameras are great to highlight heat emitted by said defective parts, as well as monitoring electronics to ensure they are not under too much heat, making the cameras extremely useful for fixing and maintaining complicated heat-sensitive electronics.  However, electronics are not the only examples that show symptoms of heat when they malfunction.  Humans also have fevers when they are sick, during which their heads would be abnormally hot.  Therefore, in the case of a pandemic, thermal cameras have also been widely implemented to effectively detect people who could be suffering from diseases, such as coronavirus, SARS, or Ebola, based on their body temperature [1].

However, the technology still faces a number of problems and deficiencies. In the case of detecting the sick from the healthy, the difference in temperature between a healthy person and one with a severe fever is only about a few degrees Celsius apart. While a lot of thermal cameras can measure temperatures ranging from -40 degrees to 550 degrees Celsius, they are not extremely accurate to do precise measurements in narrow temperature ranges, such as temperatures of the human body, where a human with a body temperature of 37 degrees Celsius is considered normal but one with 38 degrees Celsius is considered to be having a fever [2].

On the other hand, each pixel generated on the thermal image requires a rather costly thermal sensor (also known as a thermocouple).  Most consumer grade thermal cameras have extremely low resolutions.  The resolution of Teledyne FLIR’s thermal cameras are between 320 x 240 and 640 x 480 [3] due to the lack of sensors.  The low quality of the image obtained means the device can only work in close proximity in order to capture a clear image of the targeted object.  Its effectiveness plummets once the target moves a few metres away.

Moreover, thermal cameras require complicated calibrations in order to be operated as intended.  Thermal intensities of an object detected by the thermal camera varies with temperature and the distance between the object and the camera itself.  So many variables are at play which can hinder the accuracy of the readings of the thermal camera. An object too far, or even too close to the camera as intended could give a different thermal intensity reading. One common way to get an accurate reading from the camera is to calibrate the device with a heat source with known temperature, also known as a blackbody reference source, as a reference set at a fixed location[4]. Different solutions have been come up with, but most of them still have their own problems and drawbacks if not extremely costly.  Due to the camera’s lack of range, the design of the heat reference could also become an issue as the camera could not pick up the heat source if it is too small.

The recent pandemic has shown us how quickly a disease can spread without strict supervision and how much damage it can cause to our society. Developing an economically friendly thermal camera calibration set up could prove impactful for small to medium sized businesses to differentiate the sick from the healthy in a cost-effective manner. Moreover, with the simplified set up, the system can be more accessible to a wider range of users such that it could be implemented in most public areas to sever the contagion chain of infection and become the ubiquitous device that promotes public health safety by effectively preventing the spread of the pandemic.

## 1.2.  Objective

The project aims to design and develop an alternative method to calibrate thermal cameras at a longer distance and at a lower cost with the use of economical heat sensitive paint.  With its property of changing colour according to the temperature of the object it has been applied to, a large surface of paint can tell the thermal camera at a distance the temperature of the object.  By combining temperature information from the paint, and the temperature data received directly from the object, the camera can be calibrated accurately at a longer distance.  As we can tell the temperature of the heat source by the change of the heat sensitive paint colour, we can conveniently calibrate the thermal camera based by calculating the difference between the activation temperature of the pain and the detected temperature of the calibration area.

### Objective Statements

* 1. To set up a calibration area with the use of heat-sensitive paint
  2. To calibrate thermal camera by tracking colour of heat sensitive paint
  3. To track and display temperature of people at different distances

## 1.3.   Literature Review of Existing Solutions

There are several solutions in the market to calibrate a thermal camera, mainly through multi-point calibrations done by manufacturers and kiosk stands owned by individual consumers.

### Multi-Point Calibration

One of the solutions FLIR [5] has come up with to recalibrate a thermal camera is by setting an array of highly accurate blackbody reference sources of different known temperatures and having the camera point at each of them one by one with the use of a robotic arm. By obtaining multiple sample points, the thermal camera is then able to estimate the relationship between an object’s thermal intensity and its temperature by plotting an approximate curve. The recalibration process has to be done in a lab regularly in order to ensure the camera’s accuracy.

Since the detectors are not linear and due to interference with other electronics, the thermal intensity- temperature curve is not exactly a simple line [4].  However, through obtaining several samples, the camera is able to plot more points onto the model, allowing it to plot a more accurate thermal intensity curve and thus have a higher temperature reading accuracy.  Meanwhile, the camera only has to be recalibrated once for a long period of time without having to go through the process again.

However, this method is not only costly, but also extremely inconvenient. The device has to be calibrated in a controlled environment with almost no variable that would affect the calibration process. Thus, the camera cannot be recalibrated unless being carried out in a certified lab. Moreover, blackbody sources had to be controlled with extreme precision. The high-quality heat sources have to be certified and traceable back to international standards, making them extremely costly [5].

### Kiosk

An alternative solution being brought up is a kiosk. A kiosk is a blackbody reference source that is being placed at a distance from the camera during operation [6]. Unlike the previous solution where the camera has to be sent for recalibration after a long period of time, the heat source of known temperature is placed right next to the target, where they can be detected by the thermal camera such that the relationship between the intensity of IR radiation and temperature can be constantly calibrated to the correct value.  Since the reference source is being monitored by the thermal camera constantly, the camera is always being calibrated to measure the dedicated spot accurately.  This solution is a lot more consumer friendly for any person to recalibrate their camera anytime they want by themselves, tackling the inconvenience issue the previous method has.

However, now that consumers have to purchase their own black body reference source, the price of a complete set of thermal camera equipment just skyrocketed with more equipment to be handled and taken care of.  A blackbody reference could be extremely expensive as previously mentioned due to the high manufacturing requirements. Moreover, unlike a lab calibrated camera which can be independently used, the device now has to pair with the reference source whenever it is being used. Besides, this solution has limitations of its own.  The target to be measured had to stand at a fixed location next to the reference source in order to give an accurate temperature, making the range of measurement extremely narrow as a few centimeters away from the reference point could mean a massive error in the temperature reading.

### Temperature Sensor and Camera Feedback

In order to avoid having to use expensive blackbody references, people have used something as simple as an electrical warmer as reference [7].  Like the kiosk, the consumer-friendly heat source is being placed at a fixed distance from the camera while connecting it to a thermometer in order to tell the camera how hot the reference source is.  The camera can then be constantly calibrated while accurately measuring the temperatures of the intended targets. In order for the distant thermometer to communicate with the thermometer, a QR code on an E-paper display is also placed with the heat source such that the thermal camera pointing at it can scan the QR code in order to read the warmer’s temperature.  Without the use of expensive blackbody references with extreme quality, a low-cost electrical warmer is way more economical for consumers to recalibrate thermal cameras on their own.

However, given the low resolution of thermal cameras, a highly complex QR code with a large amount of black and white pixels is proven difficult for the camera to scan clearly.  And since this solution works similarly to a kiosk, the target to be measured has to be placed near the reference source such that its temperature can be measured accurately.  This greatly hinders the maximum distance the reference heat source can be placed from the camera, thus also greatly hindering the effective range of it as well.  The reference source can be placed further away if the thermometer from the heat source can communicate with the camera with a simpler and larger image, for example, a large wall of paint in a single colour that indicates the temperature of the remote heat source.

The mentioned solutions are all great attempts to recalibrate a thermal camera, but not without their flaws. In the following we shall explain how we are going to tackle this problem with heat-sensitive paint.

# SECTION 2 – METHODOLOGY

## 2.1 Overview

### 2.1.1 System Description

Our proposed thermal camera calibration setup consists of two major parts, firstly the calibration device, and secondly the thermal and colour camera and the calibration program.

For the calibration device, a heat source shall be positioned near the targeted object, for example, around the head position of an average person. The heat source shall be in full contact with a conductive sheet covered in heat sensitive paint, its heat will directly be conducted to the paint such that it has the same temperature as the heat source, acting as a simple temperature indicator for the heat source by using colour. However, as one HSP is only to change from one colour into another when one temperature is reached, it can only act as a binary indicator. Hence, we would have multiple HSPs on the calibration area to indicate the temperature range of our heat source. Our RGB camera can then obtain such data by continuously observing the colour change of the multiple HSPs and reading their colour values. Although the colour change of the HSPs are distinct in colour and brightness, to avoid any unseen circumstances which would alter the visual data of the HSPs, i.e. a hairdryer- like massive heat source that would alter the temperature of our intended heat source and HSP, or a reflected light source that would alter the brightness values of our HSPs, we had implemented a dark HSP with its colour-changing point being higher than the maximum temperature of our heat source, acting as a debug, such that when this HSP in particular has its brightness value higher than a specific value, we deduce that an unexpected event has occurred and thus the data we obtained at that moment was inaccurate and should be ignored.

On the other hand, the camera set up consists of a thermal camera as well as an RGB camera. The two cameras are positioned as close together as possible such that they can capture a similar image. The RGB camera is used to reading the HSP brightness values to deduce the temperature of the heat source as mentioned, while the thermal camera is responsible for reading the temperature of both the intended target as well as the calibrator heat source.

The RGB camera would locate the calibrator and continuously monitor it. It will then capture the exact frame when the HSP changes colour and will communicate with the thermal camera to do the same, capturing the temperature of the pixel representing the calibrator. Based on which HSP has changed colour at that frame the calibration program will deduce the temperature of the heat source, i.e. If the greenish blue HSP turns yellow, we would deduce that the heat source is currently at 35oC as that is the colour changing point of the paint.

The thermal camera would initially approximate the temperature of each captured pixel based on the temperature of infrared radiation received and preprogrammed data. However, the approximation curve was inaccurate. To recalibrate the thermal camera, we would readjust the calibration curve linearly based on the difference in temperature reading on the calibrator and the thermal camera. For example, the RGB camera recorded the calibrator to be 35oC, but the thermal camera determined the calibrator to be 20oC, the difference of the temperature measured is +15oC, hence we would adjust all the temperature data of every pixels of the thermal image to increase by 15oC.

### 2.1.2 System Block Diagram

**Diagram

Description automatically generated**

###### Figure (1): Thermal camera calibration by heat sensitive paint and temperature measurements.

### 2.1.3 Component List

|  |  |
| --- | --- |
| **Items** | **Specification / Model** |
| Raspberry Pi | Raspberry Pi 3B6 |
| Heat sensitive (thermochromic) paint | Colour changing range of 30 to 40°C |
| Heat Source | single-use warmer pack |
| Thermal camera | MLX90641 55o FOV |
| Colour camera | Logitech C310HD Webcam |

Table (1): List of components

### 2.1.4 ECE Knowledge

Some knowledge gained from the ECE requirement or elective courses will be applied to the project, the courses will provide us different ideas and approaches to different parts of our FYP. For example:

* COMP1021 Introduction to Computer Science  
  This course covers the fundamental of computer programming, especially in Python, which will be our main programming language for the face-tracking and calibration system on the cameras and display.
* ELEC1100 Introduction to Electro-Robot Design  
  This course introduces the fundamental knowledge on the design, implementation and evaluation of a robot and its systems, which includes using an Arduino microcontroller to control electronic devices, and an introduction to PWM signals. This will prove necessary as we are considering using an Arduino microcontroller to control the calibration heat source should we approach the project using Peltier elements, since they will be able to be more easily controlled using a PWM signal and the microcontroller will allow us to control the feedback loop.
* COMP2011 Programming with C++  
  This course introduces the C++ programming language and a bit of Object-oriented programming, which will prove useful in programming the cameras and the feedback system on the Arduino.
* ELEC2350 Introduction to Computer Organization and Design  
  This course covers topics related to computer hardware and software organization, including computing systems, programming, and hardware-software collaboration, which help us develop the interface between our hardware and software parts and help in debugging.
* ELEC4240 Deep Learning in Computer Vision  
  This course covers the basics and various applications of deep learning and machine learning in computer vision, which will greatly help our endeavours to apply face-tracking onto the colour camera.
* ELEC4310 Machine Learning on Images  
  This course introduces methods to process images on a computer and implementation of deep neural networks, which will be useful in processing the images or video from the cameras to find the calibration plane and to track faces while filtering out other noise in the captured images.

## 2.2 Objective Statement Execution

### 2.2.1 Setting up the calibration area with Heat sensitive paint

#### Task 1: Testing colour changing properties of the heat sensitive paint

Group Member in Charge: LO, Yat Hei

Firstly, the heat sensitive paint that we choose should have a colour changing temperature range that is near the human body temperature, i.e., 30 – 40°C. Secondly, the heat sensitive paint should have a measurable difference in colour when the temperature is varying and said colour change should be able to be captured by a colour camera clearly.

We have purchased several heat sensitive paint with a colour changing temperature of 35, 38 and 45 degrees Celsius respectively. During our feasibility testing, the paint is painted on a piece of tinfoil, owing to its conductive nature. The paint show a clear colour change, with the bluish green paint turning yellow swiftly when it reaches a temperature of 35 degree Celsius, the dark blue paint turns light blue when the temperature reaches 38 degree Celsius, and finally, the dark brown paint turns bright red gradually when a temperature of 45 degree Celsius or higher has been reached.

The observed colour change was quick and happens at around the same temperature every time, as measured with an infra-red temperature sensor. The colour changing properties of the heat sensitive paint does not change either after being exposed to air under long periods of time, the heat sensitive paint was still able to change colour instantly despite being dry for several months during our testing.

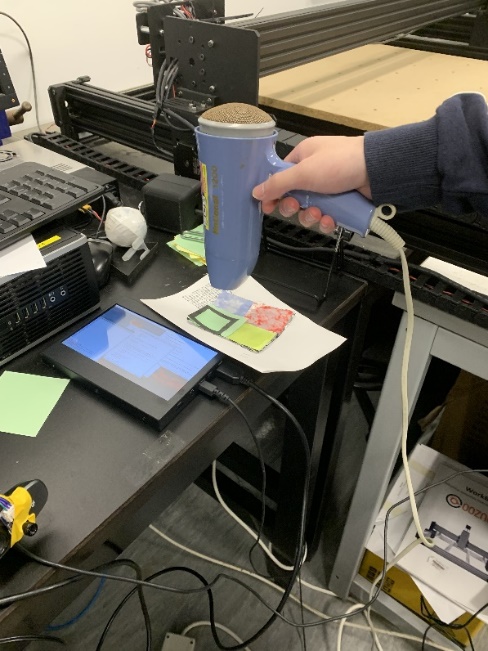


Figure (2): Result of 3 HSP turning into brighter colours under extreme heat with a hair dryer.

#### Task 2: Mapping the colour changes of heat sensitive paint to different temperatures

Group Member in Charge: LI, Hong Yin

In order to be able to determine the temperature of the calibration area accurately, we will have to map the RGB values of the heat sensitive paint captured by the colour camera to different temperatures to create a reference chart to calibrate the thermal camera with.

We have recorded the colour change of the heat sensitive paint and have found it to be a sharp change between a dark and light colour. Therefore, the camera would not have to recognize all three colour values of the heat sensitive paint to detect its colour change, instead it only has to identify one of the values. Given the colour of the three paints closely resemble the three primary colours, our program reads different values for each of the colours, i.e. The green paint turns yellow at 35oC, hence it’s G value changes more drastically than the other 2 values. On the other hand, we read the R value of the dark brown paint to tell when it turns bright red when it reaches 45 oC. Finally, we check the B value of the dark blue paint to check whether it turns bright blue when a temperature of 38oC has been reached.

#### Task 3: Testing different electric heaters or hand warmers

Group Member in Charge: LO, Yat Hei

The temperature of the calibration area should be within 30 – 40°C to provide a reference temperature for the thermal camera. Therefore, we need to test different ways of heating up the calibration area uniformly and test the effects of environmental factors such as air flow.

We have opted for a simple single-use warmer pack that has a relatively large surface and heats up to around 40 degrees, which is within the temperature range of our 3 HSPs. While a new warm pack has to be used every time, it can stay heated for long periods of time for our calibrator to work. The specific heat pack we used had a measured average time to stay heated for 2 hours.

An electrical warmer was considered as a substitute as their temperatures can be controlled and coarsely adjusted for our purpose. However, a lot of the electrical warmers we found online could not run very long, with some only being able to run for less than an hour before having to be recharged again. Moreover, we would like to make our calibration setup as simple as possible and avoid the use of electrical devices on our calibrator.

Given that the single-use warmer packs are sufficient for our cause and that it does not have any significant drawbacks, we opted for the heat packs for its lower cost as well as its ability to work without a power source, allowing our calibrator to be a simple as possible.

|  |  |  |
| --- | --- | --- |
|  | One-Use Warmer Pack | Electrical Warmer |
| Advantages | -No electricity needed  - Long run time(~2 hours or more) | -Usually has around 3 settings for temperature |
| Disadvantages | -No temperature control  -Can’t be reused | -Needs charging/battery replacement  -An average run time of less than 1 hour |

Table (2): Comparison between one-use warm packs and electrical warmers

#### Task 4: Creating an enclosure for the calibration area

Group Member in Charge: LO, Yat Hei

The calibration area, including the heater needs to be housed in a housing of some sort, and should be able to either be attached to a wall or be free-standing.

Our enclosure is a simple stand of around 1.5 meters made of wood with a wooden frame to hold our aluminium foiled HSP calibrator as well as our heat pack. The use of a stand would allow us to relocate our calibrator in different locations and distances to suit the needs of the situation, giving us more flexibility.

#### Task 5: Designing a layout for the calibration area

Group Member in Charge: LO, Yat Hei

A layout of the calibration has to be designed for the camera to collect data from the HSP. The layout must be simple and large enough that the camera can distinguish the pixels of the HSP on the calibration area.

The calibration area was made of a sheet of aluminium foil for its heat conducting properties with its layout being divided into 4 squares of equal size, having the QR pattern used for tracking on the top left corner of the calibration area, the greenish blue 35 oC HSP located on the bottom left, the dark blue 38 oC HSP on the top left, and the 45 oC dark brown HSP locating on the bottom right corner respectively. The size of the calibration area overall is approximately the size of our handwarmer, allowing heat of the warmer to conduct to the 3 HSPs as evenly as possible with the warmer being clamped behind of the calibration area, such that the temperature data sent by the calibrator could be consistent.

A picture containing indoor, ceiling, floor

Description automatically generated

###### Figure (3): Setup of our HSP panel

#### Task 6: Evaluation of setting up calibration area:

Expected Outcome:

To create a reliable calibration area mainly composed of heat sensitive paint.

Actual Outcome:

The current layout of heat sensitive paint has proven to work as intended. The camera is able to identify the colour values of the HSPs and is able to detect the moment the paints change colour. However, if more heat sensitive paint with a thinner colour-changing temperature gap could be utilized, the temperature data provided by the various paint would be more precise and accurate for our calibration algorithm.

### 2.2.2 Calibrating thermal camera by tracking colour of heat sensitive paint

#### Task 1: Testing thermal and colour camera

Group Member in Charge: LI, Hong Yin

The field of view and frame rate of the colour camera should be larger and higher than that of the thermal camera. Also make sure that the captured video output should be able to be shown on the computer display.

We are currently using the MLX90641 thermal camera connected to a Raspberry Pi 3B, and sending the captured data through Wi-Fi to a laptop, where it is shown in a cv2 window. The colour camera we are using is the external Logitech C310HD that connects to the laptop through a USB cable, also shown in a cv2 window.

The thermal camera however, despite being very affordable, has an extremely low resolution and framerate, making it very hard to capture detailed moving objects. The thermal camera has an average of 0.8 fps and 24x32 pixels and the delay between capturing and outputting images is rather noticeable. It is sometimes difficult for us to run tests due to the specs of the thermal camera, such as matching the thermal camera and the RGB camera images, which will be mentioned in the following. However, for the purpose of a low-cost thermal camera calibration setup, the thermal camera is sufficient for our cause.

A picture containing text, computer, electronics, desk

Description automatically generated

###### Figure (4): Image output of the thermal camera

showing one of our groupmates with very low resolution

The RGB camera on the other hand, we were using a Logitech C310HD webcam with 720p resolution, 30 fps and 60o fov. With the thermal camera being the limiting factor of our setup, the specifications of the webcam is sufficient for our cause. On the other hand, the USB connected webcam allows it to be adjusted easily at different angles and positions for calibration. Meanwhile, it can be easily reattached and shared across different groupmates to test out different parts of the calibration program. Its high specs, flexibility and ease of use made it our primary RGB camera choice.

A picture containing electronics, camera, black

Description automatically generated

###### Figure (5): Logitech C310HD Webcam

#### Task 2: Field of view of thermal camera and colour camera

Group Member in Charge: LI, Hong Yin

The field of view of view of the colour camera should be cropped and mapped to that of the thermal camera to facilitate easier information transfer between cameras in the calibration and tracking programs.

We have lined up and matched the positions of thermal and colour camera on the webcam and are able to find the position shown on the colour camera on the thermal camera with relatively good accuracy.

A picture containing indoor, electronics

Description automatically generated

###### Figure (6): Placement of thermal camera right next to the webcam

As mentioned, the field of view of the 2 cameras where different, with the thermal camera having 55o fov whereas the RGB camera has 60 o fov. The RGB camera could capture images at a wider angle while the thermal camera tends the stretch the image such that objects in the image becomes a bit taller.

We have determined the field of view of the 2 cameras by moving an object around until it reaches the edge of the image. We then cropped the output images accordingly and shifted the thermal image lower such that it matches the greyscaled image as much as possible.

That said, while the 2 cameras were aligned such that both images of an object placed at less than 2 meters away from the cameras can be precisely overlapped, the thermal image can be seen diverting away from the RGB image to the left for objects at 3.5 meters away from the camera, which introduces inaccuracy to the set up at further distances.

A screenshot of a video game

Description automatically generated A screenshot of a video game

Description automatically generated

###### Figure (7) & (8): Thermal Image and RGB image of a person standing at 3.5m away.

While the face detection program for the RGB image works accurately, the face tracking data directly translated into the thermal image was slanted to the right side of the person.

#### Task 3: Finding the calibration area

Group Member in Charge: LI, Hoi Him

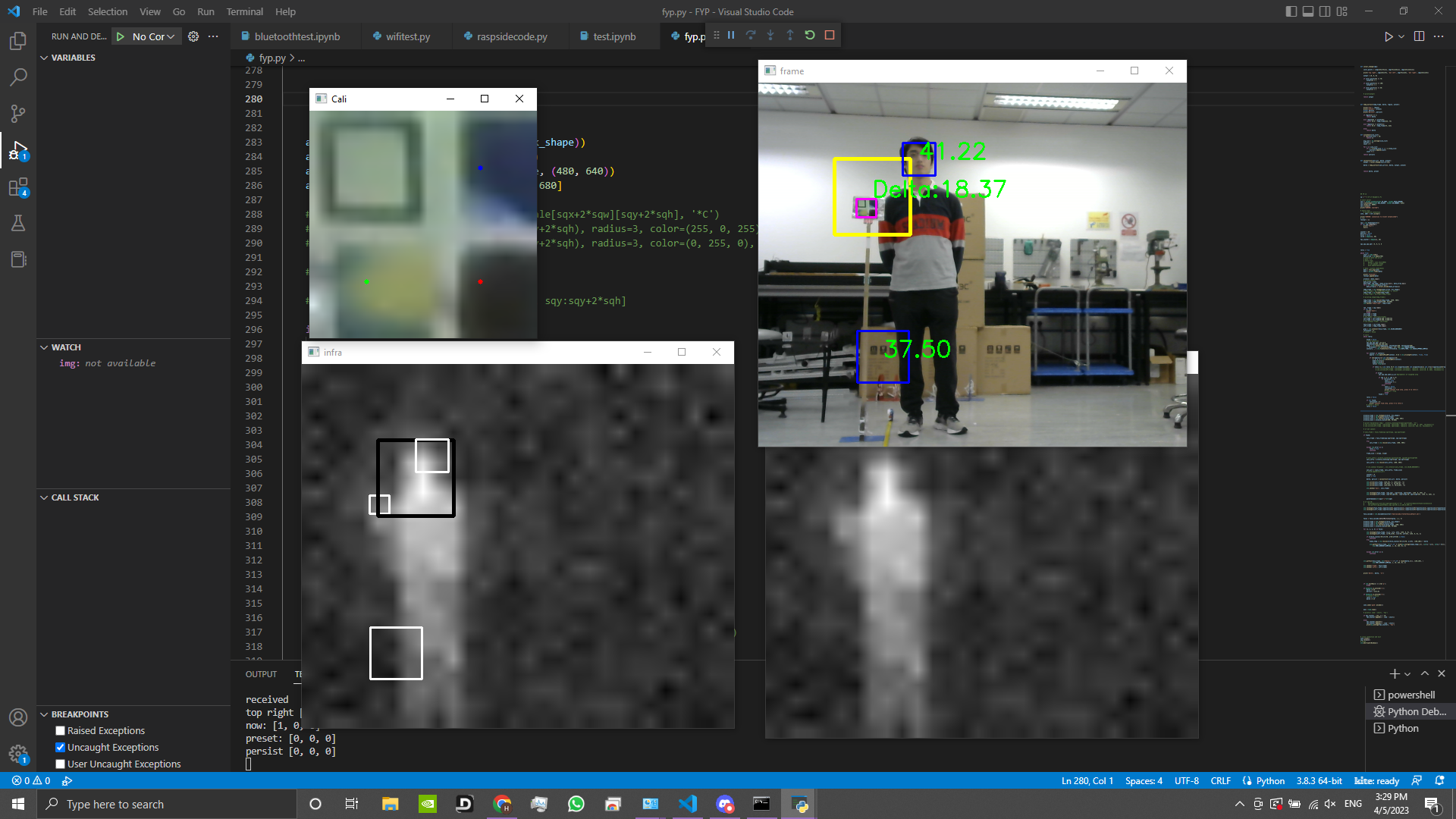
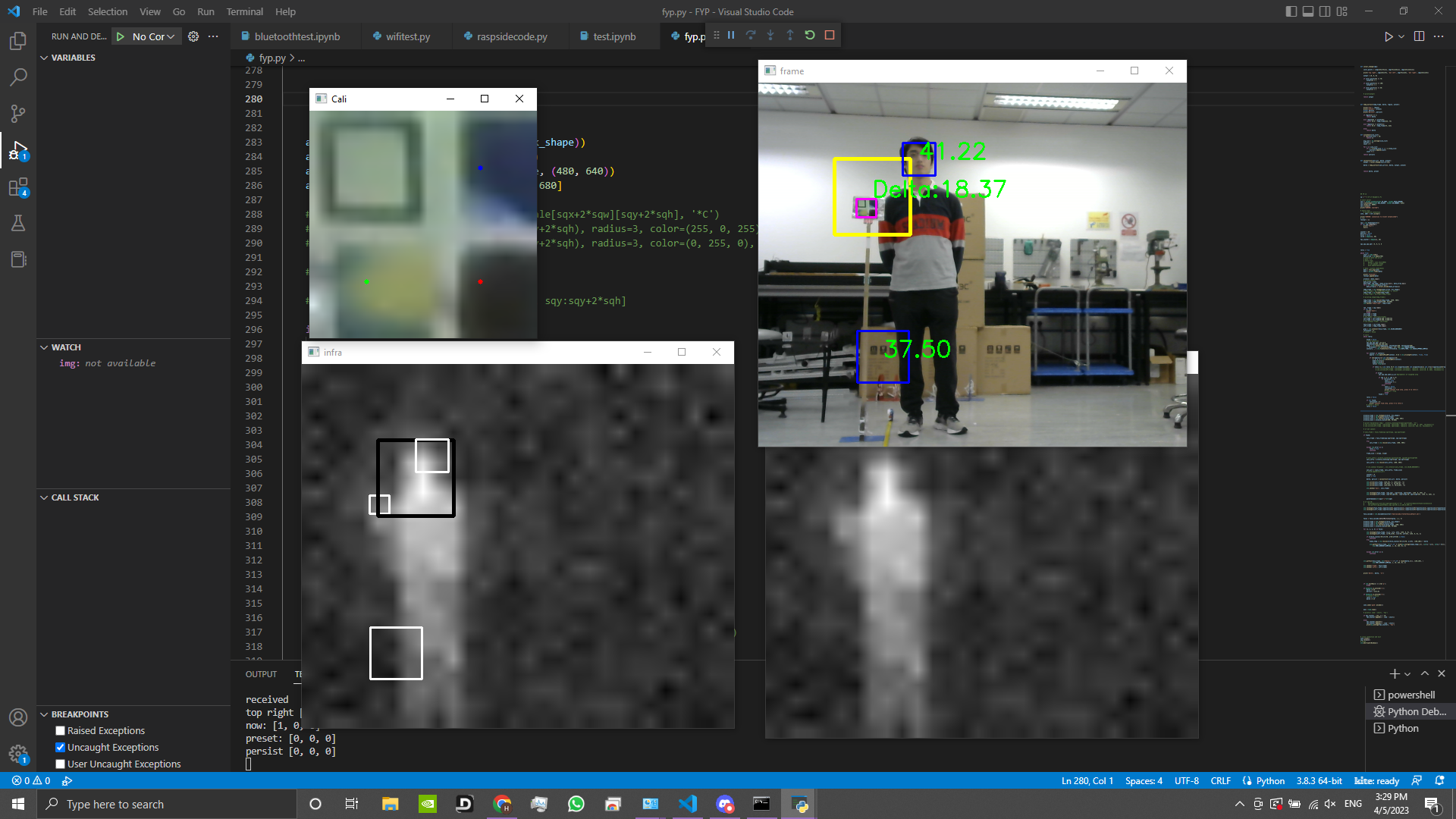
The calibration area should be able to be found by the colour camera and its location shared with the thermal camera output for calibration.

For the RGB camera to locate the calibration area, we’ve made a custom pattern for the camera to identify using image processing. The pattern consists of a large bright square background and a square of the same colour with a thick black border. This pattern is then placed on the top left corner of the calibrator, taking up 25% of the space of the calibrator.

Our program primarily utilizes contour detection to locate the calibration area. First, a specific part of the image window obtained by the camera is boxed up, the program will only look for the QR pattern in this specific area to reduce noise in the form of similar patterns, which reduces cases of tracking unrelated objects, as well as reducing the workload of the program as it only has to process a specific area in the image instead of all 1280x720 pixels, hence increasing its speed as well.

Then, the program looks for rectangles with a width-height ratio of 0.9-1.1, which are the dimensions of a square. Given that the program sees the bordered square as two squares, as a thickened square border has both an inner square contour and an outer square border, the program only identifies the larger square as our intended target if there were two squares with the centre at the same coordinates, implying that it is a thick bordered square.

Finally, given the assumption of the width and height of the QR pattern being half of the calibration area respectively, the program boxes up the area 4 times of that of the QR pattern, with the coordinates of its top left corner being the same of that of the QR pattern, hence locating the area of the calibration area.

###### Figure (9): Example of program indicating the tracking area

in yellow borderd square and marking the calibration area located 3.5m away from the RGB camera with a pinksquare

###### Figure (10): Cropped image of said designated calibration area

With the design of this QR pattern, the camera can track the calibration area continuously up to 3 meters distance. Beyond the 3 meter limit, the camera starts to have trouble identifying the QR pattern as the square border becomes so small and thin the two square contours start to overlap each other, hence not being picked up by the RGB camera.

To tackle the issue, we also came up with a manual reset button that programs the RGB camera to only track the calibration area once, with a manual refresh button. Assuming that the calibration area stays stationary upon set up, the RGB camera only tries to look for the calibration area once upon refresh and assumes the area to be our calibration area from then on. Our calibration program will then work with the colour values within this marked area for calibration.

#### Task 4: Heat sensitive paint colour calibration

Group Member in Charge: LI, Hoi Him

The goal of this step is to make sure that the colour of the heat sensitive paint needs to be zeroed in with the known RGB values of the colour strips around the calibration area to remove any bias and errors due to environmental factors such as lighting and shadows.

As mentioned previously, the colour change is distinct and easy to determine due to the colour change between a dark colour and a light colour at the paint’s activation temperature; hence not requiring exact RGB values for the camera to recognize the colour change.

We have chosen three different heat sensitive paints to create the calibration plane. They are 35°C – Blue to Yellowish-green, 38°C – Dark blue to light blue, 45°C – Dark brown to Red respectively. After testing different filters, we have determined to use different colours filters to determine the colour change. For the 35°C heat sensitive paint, the change is determined at 70 in the red RGB value, the 38°C paint at 100 in the green RGB value, and the 45°C paint at 50 in the red RGB value.

#### Task 5: Evaluation of tracking colour of heat sensitive paint

Expected Outcome:

To create an image tracking system that could reliably track the calibration area at up to 4 meters range continuously.

Actual Outcome:

The Tracker system would only reliably track the calibration area continuously and reliably up until 3 meters. Beyond 3 meters it can only identify the calibration area one-fifth of the time. Although a manual refresh backup measure has been implemented, a better QR tracking pattern of more complex shapes and the better utilization of online resources would be beneficial to enhance the effectiveness of the tracker system.

### 2.2.3 Track and display temperature of people at different distances

#### Task 1: Calibration algorithm for thermal camera

Group Member in Charge: LI, Hong Yin

The goal of this step is to develop an algorithm to be able to calibrate the thermal camera with the information provided by the colour camera, i.e., position of the calibration area, colour of heat sensitive paint translated into temperature. With the help of the datasheet and online tutorials for Python coding with the thermal camera, we should be able to develop a reliable calibration algorithm.

In our testing, we are able to extract temperature data from the thermal camera and be transmitted to the main computer. Our calibration algorithm relies on the colour camera detecting the calibration area, matching it to the thermal camera. If the colour camera detects a colour change in the calibration area, the activation temperature of the heat sensitive paint that changed colour is then compared to the temperature the thermal camera detects at the calibration area . The temperature difference ‘delta’ is then calculated with . With the temperature difference , all temperatures detected by the thermal camera is then offset by adding .

#### Task 2: Debugging and testing of calibration algorithm

Group Member in Charge: LI, Hoi Him

To test the calibration algorithm and the limits of the thermal camera, we will put the calibration area at different distances to see whether the calibration will still work.

We had tested the calibrator at different distances, from 1 meter to 4 meters by measuring the body temperature of one of our teammates with different temperature measuring equipment for comparison at various ranges in the increments of 0.5 meters. While the calibration program can accurately measure a person’s body temperature up to 2 meters distance, the program is starting to have trouble locating the calibrator at 3 meters distance, and at 4 meters distance the camera can barely see the calibrator, yet it still can be reliably calibrated once it successfully identifies the calibrator, making 4 meters the maximum effective range of our thermal camera with the limiting factor being its ability to identify the calibrator.

#### Task 3: Face tracking

Group Member in Charge: LI, Hoi Him

Using the colour camera, we should be able to track human faces and draw a rectangular bounding box around them and show it on a display screen, such as a laptop. This should be able to be done with either OpenCV or existing pretrained machine learning libraries. The bounding box location should be shared with the output of the thermal camera to capture the location of where temperatures need to be taken.

The face tracking is currently done with acceptable accuracy on the RGB camera, and with the coordinates of the face extracted and put into the frame of the thermal camera output, we are able to draw bounding boxes around faces in both the colour and thermal camera output frame. We plan to improve the accuracy further since sometimes it will take a while for the program to recognize people with different shaped masks.

The face tracking program can detect multiple faces at once as well, allowing the possibility for our program to measure the body temperatures of multiple targets at once. That said, with the limited window size of our setup, it is only possible to fit in a maximum of two people simultaneously within the visible range of our thermal camera.

#### Task 5: Evaluation of track and display temperature of people at different distances

Expected Outcome:

To create a calibration system being able to reliably track people in sight and read their body temperature at a maximum of 4 meters range and with an accuracy of +-1 oC and precision of 2 decimal places.

Actual Outcome:

While the camera is able to track targets facing the camera at 1 meter range and deduce their body temperatures with an error of 0.31 oC, exceeding our expectations, the accuracy quickly drops, and by 2 meters of distance, the camera is already having an error of 1.81 oC. Furthermore, with the limitations of resolution of our thermal camera, the reliability of our calibration system plummets beyond the 2-meter range, having an error of 3.42 oC. While the statistics were small it is proven significant as a 0.1 oC difference is all it takes to differentiate a healthy person from a heavily ill patient.

Although the effective range of the calibration system was halved and the error rate was double from expected, given how the calibration had improved the accuracy of the MLX90641 infrared camera by an average of 84.5%, the results were still acceptable.

One way to improve would be to implement the calibration system on an infrared camera with higher resolution to resolve objects at higher definition for better tracking as well as use a more convoluted algorithm to adjust the calibration curve and better fit the realistic situation instead of a simple linear one.

### 2.2.4 Displaying final results

#### Task 1: Temperature from thermal camera

Group Member in Charge: LI, Hong Yin

The program has to output its processed data in a effective manner to satisfy its primary objective.

To achieve said task we decided to have a person’s body temperature data placed near the corresponding box that is tracking the targeted person. The data in bright green for high contrast would follow the box at a relative position to indicate which person’s body temperature it relates to. The task was relatively simple and straight forward and the data can be effectively presented on the output window of our program.

#### Task 2: Showing temperature and bounding box in display

Group Member in Charge: LI, Hoi Him

The face tracking provides a bounding box and the coordinates to place the person’s corresponding temperature as text next to them. The temperatures can be extracted as the frames of the thermal camera and colour camera has been manually calibrated to match each other with decent accuracy from 1 to 4 meters. Using cv2, the face is tracked with a bounding box with the detected temperature displayed in text next to the box.

We have tested the setup from 1 to 4 meters, with varying degrees of accuracy. The cameras are capable of recognizing the calibration area and faces along with their temperatures being shown next to the faces.

#### Task 3: Evaluation of displaying final results

Expected Outcome:

Create a display system for users to easily ready the temperature data intended for each target in front of the thermal camera.

Actual Outcome:

Users can easily identify tracked targets by observing the blue boxes bounded to each tracked targets’ heads, indicating that the camera identifies that object as a person and is currently reading their temperature. On the other hand, the temperature data for each person follows its intended target’s head position so users can easily identify the body temperature of a target without mixing up with other visible pedestrians in sight. All the visible data will disappear once the target is out of sight to avoid confusion.

All in all, although the detection system suffers from varying success some of the time due to various variables, the display system itself works as intended flawlessly.

### 2.2.5 Code

#### Laptop code

import socket

import numpy as np

import cv2

import pickle

import struct

import matplotlib.pyplot as plt

import time

from collections import deque

from decimal import \*

import keyboard

mlx\_shape = (24,32)

global base\_temp

global top\_temp

# temperature delta

delta = 0.0

# colour persistence

persist = [0, 0, 0]

def colour\_encode(temp):

  step = 1 / (top\_temp - base\_temp)

  colour = (temp - base\_temp) \* step

  return colour

The code above imports the libraries we will be using and defines the shape of the thermal camera (MLX90641) output that will be received from the Raspberry Pi. ‘delta’ is defined as the temperature offset used in the calibration. The ‘persist’ list is the current state of the calibration area. The ‘colour\_encode’ function is used to generate a readable image from the infra-red temperature data.

def recv\_msg(sock):

    # Read message length and unpack it into an integer

    raw\_msglen = recvall(sock, 4)

    if not raw\_msglen:

        return None

    msglen = struct.unpack('>I', raw\_msglen)[0]

    # Read the message data

    return recvall(sock, msglen)

def recvall(sock, n):

    # Helper function to recv n bytes or return None if EOF is hit

    data = bytearray()

    while len(data) < n:

        packet = sock.recv(n - len(data))

        if not packet:

            return None

        data.extend(packet)

    return data

The ‘recv\_msg’ function is responsible for the receiving of the pickled message from the Raspberry Pi through the Wi-Fi socket. The ‘recvall’ function is a helper function that makes sure the entire message is received before unpacking it into a numpy array, and avoid any errors with truncated pickle data.

def colour\_change(img):

    colour\_points = [img[225][75][2], img[75][226][1], img[225][225][2]]

    print('top right', img[225][75], 'bot left', img[75][225], 'bot right', img[225][225])

    output = [0, 0, 0]

    if colour\_points[0] >= 70:

        output[0] = 1

    if colour\_points[1] >= 100:

        output[1] = 1

    if colour\_points[2] >= 50:

        output[2] = 1

    # print(output)

    return output

Taking the calibration frame resized to (300, 300) as the input , the ‘colour\_change’ function is responsible for determining if the different heat sensitive paints on the calibration area have changed colours or not. Note that different heat sensitive paints use different parts of their RGB to determine their colour change., The ‘output’ list represents the states of the different paints, with ‘0’ being the lower temperature colour, and ‘1’ being the higher temperature colour.

def temp\_extract(temp\_frame, delta, region, preset):

    print("now:", region)

    print("preset:", preset)

    global persist

    print("persist", persist)

    if region[2] == 1:

        return delta

    elif region[0] != preset[0]:

        return 38.0 - temp\_frame[225, 75]

    elif region[1] != preset[1]:

        return 35.0 - temp\_frame[75, 225]

    else:

        return delta

The ‘temp\_extract’ function takes in the input of the current delta (), ‘region’ is the output of the previous ‘colour\_change’ function, and preset is the current ‘persist’ variable, which represents the last state of the calibration frame. The function compares the ‘region’ list with the ‘preset’ list, and calculates the delta should there be a colour change detected (region) when compared to the previous state (persist), returning delta as the calibrated offset.

def caliextract(cali\_arr, delta, preset):

    output = colour\_change(cali\_arr[0])

    delta = temp\_extract(cali\_arr[1], delta, output, preset)

    return delta, preset

The ‘caliextract’ function is a helper function which takes the information of the frames and calls the aforementioned functions for calibration (delta) and updating the states of the colours of the heat sensitive paints on the calibration frame (preset). The input ‘cali\_arr’ contains the captured calibration frame in both colour and temperature data.

ip = "" # IP of Raspberry Pi

# start server

serv = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

serv.setsockopt(socket.SOL\_SOCKET, socket.SO\_SNDBUF, 8192)

serv.bind((ip, 8080))

serv.listen()

print("SERVER: started")

# while True:

    # establish connection

conn, addr = serv.accept()

print("SERVER: connection to Client established")

i = 0

receipt = []

cap = cv2.VideoCapture(1)

if not cap.isOpened():

    print('fail')

    exit()

fps\_counter = deque([], 10)

sqx,sqy,sqw,sqh = 0, 0, 0, 0

This is the initialization code for the server and the RGB camera capture, and some variables that will be used later. We used the cv2 library since it has easy to use features for camera capture and integration with numpy arrays. The server is hosted by the laptop Wi-Fi hotspot to which the Raspberry Pi is connected, providing a stable connection between the two devices, and ensures that the entire message is reliably sent.

while True:

    start = time.time()

    data\_array = np.empty(768)

    data = recv\_msg(conn)

    data = pickle.loads(data)

    print('received')

    receipt.append(data)

    archive = data.copy()

    # print(archive)

    data\_array = data

    base\_temp, top\_temp = data\_array.min(), data\_array.max()

    for i in range(len(data\_array)):

        data\_array[i] = colour\_encode(data\_array[i])

    temp\_frame = (np.reshape(data\_array, mlx\_shape))

    temp\_frame = np.flipud(temp\_frame)

    temp\_frame = cv2.resize(temp\_frame, (640, 480))

    temp\_frame = temp\_frame[0:480, 70:680]

    cv2.imshow("temp\_frame", temp\_frame)

This is the start of the while loop, and initiates variables to be used in the same loop. The code starts with the receiving of the data from the thermal camera, and loads it into a variable, and have a copy created to be used later. The temperature data is then encoded into a binary image, flipped to the correct orientation, resized and cut into the correct resolution and used for display.

    ret, frame = cap.read()

    if not ret:

        print('crap')

        break

    cut\_frame = frame

    ori\_frame = frame

    cut\_frame = cut\_frame[0:480, 0:640-75]

    ori\_frame = ori\_frame[0:480, 0:640-75]

    # cv2.imshow('frame', cut\_frame)

    face\_frame = cut\_frame.copy()

    infr\_frame = temp\_frame.copy()

    gray = cv2.cvtColour(face\_frame, cv2.COLOUR\_BGR2GRAY)

    retrycount = 0

The colour camera then captures a frame, copied, and cut to match the size and resolution of the thermal camera. A grayscale image is also created to be utilized later.

# start

    while retry:

        found = False

        # area detection

        sqx,sqy,sqw,sqh = 0,0,0,0

        squaresize=[100,100,100,100]

        edges = cv2.Canny(image=gray, threshold1=100, threshold2=200)

        \_, threshold = cv2.threshold(edges, 127, 255, cv2.THRESH\_BINARY)

        contours, \_ = cv2.findContours(threshold, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

        for contour in contours:

            approx = cv2.approxPolyDP(contour, 0.01 \* cv2.arcLength(contour, True), True)

            if len(approx)>=4 and len(approx)<=8:

                x, y, w, h = cv2.boundingRect(contour)

                midx=int(x+w/2)

                midy=int(y+h/2)

                ratio= float(w)/h

                if ratio <1.1 and ratio >0.8 and x>squaresize[0] and y>squaresize[1] and (x+w)<(squaresize[0]+squaresize[2]) and (y+h)<(squaresize[1]+squaresize[3]):

                    if w>sqw:

                        sqx,sqy,sqw,sqh=x,y,w,h #parameters of targeted area

                        if sqw == 0 or sqh == 0:

                            if retrycount < 5:

                                retry = True

                                retrycount += 1

                                continue

                            else:

                                retry = False

                                retrycount = 0

                                print('cannnot find area, press E to retry')

                                # cannot = True

                                break

                        found = True

        retry = False

        if not found:

            retrycount = 0

            print('cannnot find area, press E to retry')

        # cannot = False

        retry = False

This part of the code is for the finding and recognition of the calibration area. In order to prevent any problems such as obstruction of the calibration area and to improve the stability and accuracy of the calibration, we have decided to only find the calibration frame at the start and lock it in place. Should the calibration area be obstructed or missing from the target area when the program is started, it will send a message to the user who can manually adjust the position of the calibration frame or clear any obstruction. Pressing the key ‘E’ forces the program to relocate the calibration area, and ensures that the correct area has been captured.

    archive\_frame = (np.reshape(archive, mlx\_shape))

    archive\_frame = np.flipud(archive\_frame)

    archive\_scale = cv2.resize(archive\_frame, (480, 640))

    archive\_scale = archive\_scale[0:480, 70:680]

    if found:

        cali\_frame = face\_frame[sqy:sqy+2\*sqh, sqx:sqx+2\*sqw]

        try:

            cali\_frame = cv2.resize(cali\_frame, (300, 300))

        except cv2.error as e:

            retry = True

            continue

        frame\_size = (2\*sqw, 2\*sqh)

        cali\_infra = archive\_scale[sqy:sqy+2\*sqh, sqx:sqx+2\*sqw]

        cali\_infra = cv2.resize(cali\_infra, (300, 300))

        cali\_arr = [cali\_frame, cali\_infra, frame\_size]

        counter = 0

        exist = True

This part of the code extracts the calibration area if it is detected in the previous step, and prepares it for further manipulation. With try excepts catching exceptions that might be sent due to calibration area not being detected as a failsafe for the code before, ensuring the code executes smoothly later.

        delta, persist = caliextract(cali\_arr, delta, persist)

        cv2.circle(cali\_frame, (225,75), 3, (255,0,0), -1)

        cv2.circle(cali\_frame, (75,225), 3, (0,255,0), -1)

        cv2.circle(cali\_frame, (225,225), 3, (0,0,255), -1)

        cv2.imshow('Cali', cali\_frame)

        cv2.rectangle(face\_frame, (sqx,sqy), (sqx+2\*sqw, sqy+2\*sqh), (255, 0, 255), 2)

        cv2.rectangle(infr\_frame, (sqx-40,sqy+20), (sqx+2\*sqw-40, sqy+2\*sqh+20), (255, 0, 255), 2)

    cv2.rectangle(face\_frame,(squaresize[0],squaresize[1]),(squaresize[0]+squaresize[2],squaresize[1]+squaresize[3]),(0,255,255),3)

    cv2.rectangle(infr\_frame,(squaresize[0],squaresize[1]),(squaresize[0]+squaresize[2],squaresize[1]+squaresize[3]),(0,255,255),3)

The ‘caliextract’ function is then called to update delta and persist, while creating a frame of reference for the user to observe; and should they be unhappy with the calibration area captured being too small, inaccurate, or in the wrong location, they can call for the program to find it again.

 face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

    faces = face\_cascade.detectMultiScale(gray, 1.1, 4)

    archive\_frame = (np.reshape(archive, mlx\_shape))

    archive\_frame = np.flipud(archive\_frame)

    archive\_scale = cv2.resize(archive\_frame, (480, 640))

    archive\_scale = archive\_scale[0:480, 70:680]

    for (x, y, w, h) in faces:

        cv2.rectangle(face\_frame, (x,y), (x+w, y+h), (255, 0, 0), 2)

        cv2.rectangle(infr\_frame, (x-40,y+20), (x+w-40, y+h+20), (255, 0, 0), 2)

        if archive\_scale[x-40:x+w-40, y+20:y+h+20] is None:

            continue

        try:

            human\_temp = cv2.resize(archive\_scale[x-40:x+w-40, y:y+h], (100,100)) + delta

            cv2.putText(face\_frame, str("%.2f" % round(np.average(human\_temp),2)), (int(x + w/2), int(y + h/2)), \

                cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)

        except cv2.error as e:

            continue

This is the start of the face recognition part of the code. The RGB camera frame taken from the backup is first gray-scaled, then sent for face recognition. The for-loop draws bounding boxes around all found faces in both the RGB camera and thermal camera windows. Since ‘faces’ contains all the coordinates of the detected faces, the coordinates are translated to that of the thermal camera and the temperatures extracted. The temperature is then averaged and displayed next to the face in the colour camera frame, with a try except to catch any exceptions that might be thrown due to the frame being a line or None.

 cv2.putText(face\_frame, str("Delta:" + ("%.2f" % round(delta,2))), (150,150), \

                cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)

    cv2.imshow('frame', face\_frame)

    cv2.imshow('infra', infr\_frame)

    print('Delta', delta, '\*C')

This part of the code shows the current delta next to the calibration area on the colour camera frame and calls imshow for both the temperature and colour frames.

if cv2.waitKey(1) == ord('q'):

        break

    if keyboard.is\_pressed('w'):

        delta = 0.0

        persist = [0,0,0]

    if keyboard.is\_pressed('e'):

        # cannot = False

        retry = True

        delta = 0.0

    conn.send('good'.encode())

    end = time.time()

    # print(1/ (end - start), 'fps')

    if fps\_counter.\_\_len\_\_() < 10:

        fps\_counter.append(1 / (end - start))

    else:

        fps\_counter.popleft()

        fps\_counter.append(1 / (end - start))

        print(np.average(fps\_counter), 'fps')

# close connection and exit

conn.close()

cap.release()

# break

cv2.destroyAllWindows()

The first if statement ensures that the user can manually end the program when needed by pressign the key ‘q’. The second if statement is for debugging and resetting the delta and current persist variable by pressing ‘w’. The third if statement forces the program to relocate the calibration area and sets the delta to 0.

The program then sends a recognition message to the Raspberry Pi and signals for the next frame to complete the loop. The start and end time is taken to take an average frames-per-second of the program, which fluctuates around 3.2fps most of the time with the thermal camera running at 4Hz.

#### Raspberry Pi code

By modifying the code provided in a tutorial by makersportal on the MLX90640, we managed to achieve a higher frames per second, and can then send the data to the laptop via Wi-Fi hotspot.

import socket

import numpy as np

import adafruit\_mlx90640

import time,board,busio

import pickle

import struct

ip = '192.169.137.1'

client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client.connect((ip, 8080))

print("Client connected")

First we import the libraries that are going to be used and set up the MLX90640 thermal camera through I2C protocol. We had to lower the max refresh rate since our Raspberry Pi could not handle the higher refresh rate limit of 16Hz, and cannot practically send a full message while running at 8Hz.

def send\_msg(s, msg):

    msg = struct.pack('>I', len(msg)) + msg.pack

    s.sendall(msg)

The send\_msg function handles the sending of the thermal camera temperature data to the computer. It pacts it into a struct and ensures that the data sent is always complete.

while True:

    i2c = busio.I2C(board.SCL, board.SDA, frequency=1000000)

    mlx = adafruit\_mlx90640.MLX90640(i2c)

    mlx.refresh\_rate = adafruit\_mlx90640.RefreshRate.REFRESH\_8\_HZ

    mlx\_shape = (24,32)

    frame = np.zeros((24\*32))

    try:

        mlx.getFrame(frame)

    except ValueError:

        continue

    msg = pickle.dumps(frame)

    send\_msg(client, msg)

    reply = client.resv(2048)

    print(reply.decode())

client.close()

For the message sending loop, the thermal camera data is captured at a rate of 8Hz, however, since the camera base resolution is 1216, the library makes sure that the camera produces a 2432 frame by halving the effective frames-per-second, ending up with 4fps being the max possible frames per second we can get. The loop also makes sure that it waits for the acknowledgement of the main computer before starting the next loop by requiring a reply.

## 2.3 Main Objective Evaluation and Discussion

Our main objective was to create a low-cost thermal calibration system with the use of heat sensitive paint that is simple to set up by anyone. This allows accurate thermal camera thermometers to become easily affordable by people and implement them in public areas to raise public health safety.

There are several parts of our projects that need to be evaluated. Firstly, the calibration area composed of heat sensitive paint needs to be seen clearly by the RGB camera such that it could clearly identify the colours of the paint. According to the results of 2.2.1, the layout of heat sensitive paint has proven to work as we expected. The camera is able to identify the colour values of the HSPs and is able to detect the moment the paints change colour.

Secondly, the image processing system. It needs to continuously track the calibrator and record the colour values that the calibrator provides. According to the results of 2.2.2, although the tracking system starts to suffer when tracking the calibration area when it was placed beyond 3 meters, the alternative back up measure was able to extend its workable range to 4 meters, though with manual input. On the other hand, the HSP calibration system was able to differentiate the change in colours of the 3 paints and deducing the temperature of the heat source for calibration.

Finally, the evaluation of the calibration algorithm. The algorithm has to create a curve to fit the data between infrared radiation intensity and temperature across a range of distances more accurately according to the real-life scenario. For this, our calibration algorithm was able to generate a linear line to readjust the IR intensity to temperature relationship. Allowing a more accurate temperature reading compared to an uncalibrated thermal camera.

To benchmark the results of our calibration algorithm, we used different temperature measuring equipment to compare their measuring results.

### MLX90641

MLX90641 is a small sized, low-cost thermal camera with 16x12 pixels of infrared array. With 4 I/O ports it can be directly implemented into a Raspberry PI’s I/O port and obtain its own measured temperature data directly from the device. The device is commonly applied in microwave ovens, movement detection systems, temperature sensor for air conditioning systems, and sometimes, high precision non-contact temperature measurements [8]. While it has 2 field of view variants, the one we were using was the 55ox 35o FOV variant. As the thermal camera we based our calibration on, it shall be used as a direct comparison between an uncalibrated thermal camera and a calibrated one.

A picture containing electronics

Description automatically generated

###### Figure (11): Image of a MLX90641

### AR320

The AR320 was a simple gun-shaped IR thermometer made by Smart Sensor designed to measure temperatures ranging from -32-380oC, with a proclaimed error range of +-2oC and a precision up to 1 decimal place [9]. The device is primarily made for detecting fault in circuit boards or other machines by detecting irregular hot spots hidden within.

A picture containing device

Description automatically generated

Figure (12): Image of the AR320 IR thermometer [9]

### Watson’s dual-mode infrared thermometer

Although only having a range of 1-3cm, the infrared thermometer is designed for measuring a person’s forehead at pointblank range. This thermometer would be a standard benchmark for it is a common household device which a majority of individuals would use to determine whether they are having a fever or not. For this type of thermometer, we took the temperature of our subject at its intended range at pointblank during each test instead of taking it at various distances.

Graphical user interface, application

Description automatically generated

###### Figure (13): Image of Watson’s dual-mode infrared thermometer

### Results

The results are as shown:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Distance (m) | MLX90641 (oC) | AR320 (oC) | Watson’s (oC) | Our Setup (oC) | Our Setup-Watson’s (oC) |
| 1 | 20.32 | 32.0 | 37.4 | 37.71 | 0.31 |
| 1.5 | 20.54 | 30.3 | 36.9 | 38.37 | 1.47 |
| 2 | 24.04 | 29.0 | 37.0 | 38.81 | 1.81 |
| 2.5 | 22.30 | 28.5 | 37.0 | 34.76 | -2.24 |
| 3 | 20.58 | 27.7 | 37.0 | 32.65 | 4.35 |
| 3.5 | 22.85 | 27.3 | 37.0 | 41.22 | 4.22 |
| 4 | 20.42 | 27.1 | 37.1 | 35.22 | -1.68 |
| Table (3): Comparison between various thermometers at different ranges | | | | | |

Chart, line chart

Description automatically generated

###### Figure (14): Comparison between various thermometers at different ranges

As shown in figure 14, the Watson’s infrared thermometer provides the most consistent temperature reading of an average of 37.05 oC. On the other hand, both prebuilt ranged thermometers did poorly with a minimum difference of 4.6 oC as close as 1 meter. The mlx did exceptionally poor with an average error of -15.5 oC (41.7%). However, with our calibration program, the mlx is able to measure the body temperature of the subject with results close to that of Watson’s infrared thermometer with 1.81 oC difference at 2m and 0.31 oC difference at 1 meter. That said, beyond 2 meters and the calibrator starts to struggle, with its temperature reading starting to fluctuate and become unstable, having an average error of 3.42 oC (9 %), yet still being more accurate than the 2 other devices. Hence the effective range of our calibrator to measure body temperatures reliably is up to 2 meters, while it can still read body temperatures at 4 meters, its accuracy becomes hindered.

In general, the development of our project has achieved our main objective. We had developed a calibrated system for thermal cameras with an effective range of around 2 meters using heat sensitive paint at a low cost. Although the system was not able to achieve an accuracy up to medical standards of +-0.1oC, the ranged thermal camera is still able to correct itself for a far more reasonable temperature value with the help of our product.

# SECTION 3 – CONCLUSION

The project aims to design and develop an alternative method to calibrate thermal cameras at a longer distance and at a lower cost with the use of economical heat sensitive paint.  With its property of changing colour according to the temperature of the object it has been applied to, a large surface of paint can tell the thermal camera at a distance the temperature of the object.  By combining temperature information from the paint, and the temperature data received directly from the object, the camera can be calibrated accurately at a longer distance. In the end, we had developed a calibrated system for thermal cameras with an effective range of around 2 meters using heat sensitive paint with minimal expenditure.

That being said, the calibration system was far from perfect. The program had a lot of flaws and places for improvement.

For the calibrator tracker system, the QR pattern and the tracking program were way too simple for the program to track. Hence, rare cases when the camera would track all kinds of other objects in the background would occur. Meanwhile, the program was having trouble tracking the calibrator beyond the 3-meter distance without manual interference. This could have been improved if we had better utilized online resources and tools such as QRtools, which is a library built for scanning QR codes. By using ready-to-run tools the tracker may perhaps run more reliably compared to a completely new program made from scratch.

For the thermal calibration, the thermal camera resolution was insufficient for calibration at longer distances. At 3 meters distance, the resolution was so low that a person’s entire head would take up only 1 pixel, with the details of the person completely gone. With the target barely visible, it was proven difficult for our calibration program to perform accurately. A better approach would be to use a more expensive camera with a higher resolution such that the calibrator has more accurate data to work with. On the other hand, our calibration algorithm readjusts the thermal camera measurement linearly, which is not the most realistic curve. Hence the thermal camera remains to be slightly inaccurate at greater distances. A more convoluted algorithm may improve the accuracy of our calibration system.

**SECTION 4 – REFERENCES**

[1] InfraredTec, “IR-based Detec­tion of Elev­ated Body Temper­ature for Coronavirus, SARS, Ebola”, [Online], Available: https://www.infratec.eu/thermography/industries-applications/medicine/detection-of-elevated-body-temperature/. [Accessed: 14th Sep, 2022]

[2] Ewa Grodzinsky & Märta Sund Levander, “Introduction to Understanding Fever and Body Temperature”, *Understanding Fever and Body Temperature*, Palgrave Macmillan, Cham, 2019, pp1-5. [Accessed: 7th Sep, 2022]

[3] Teledyne FLIR, ”Camera Resolution and Range”, [Online], Available: <https://www.flir.com/discover/marine/technologies/resolution/>. [Accessed: 7th Sep, 2022]

[4] MoviTHERM, “Calibrating a Thermal Camera”, [Online], Available: <https://movitherm.com/knowledgebase/thermal-camera-calibration/>. [Accessed: 7th Sep, 2022]

[5] Teledyne FLIR, “How Do You Calibrate a Thermal Imaging Camera?”, [Online], Available: <https://www.flir.asia/discover/professional-tools/how-do-you-calibrate-a-thermal-imaging-camera/>. [Accessed: 14th Sep, 2022]

[6] Seek Thermal, [Online], Available: <https://www.thermal.com/seekscan.html>. [Accessed: 14th Sep, 2022]

[7] Lo Kin Shing, Fok Hiu Fung, and Chan Ho Lam, *“Temperature sensor and camera feedback channel for thermal camera calibration”,* Final Year Project, HKUST, 2021. [Accessed: 10th Sep, 2022]

[8] Melexis, “Datasheet for MLX90641”, [Online], Available: <https://www.melexis.com/en/documents/documentation/datasheets/datasheet-mlx90641>. [Accessed: 10th April, 2023]

[9] Hang Yat Stationary, [Online], Available: <http://www.hangyat.com/?o=item&a=view&id=149988>. [Accessed: 10th April, 2023].

[2]: <https://link-springer-com.lib.ezproxy.hkust.edu.hk/chapter/10.1007/978-3-030-21886-7_1>

# Appendix A – Final Project Schedule

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Objective Statements** | **Task** | WK1 14/9 | WK2 21/9 | WK3 28/9 | WK4 5/10 | WK5 12/10 | WK6 19/10 | WK7 26/10 | WK8 2/11 | WK9 9/11 | WK 10 16/11 | WK 11 23/11 | WK 12 30/11 | WK 13 7/12 | WK 14 14/12 | WK 15 21/12 | WK 16 28/12 | WK 17 4/1 |
| **Setting up the calibration area** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Testing colour changing properties of the heat sensitive paint |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Mapping the colour changes of heat sensitive paint to different temperatures |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Testing different electric heaters or hand warmers |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Creating an enclosure for the calibration area |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Designing a layout for the calibration area |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Calibrating thermal camera by tracking colour of heat sensitive paint** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Testing thermal and colour camera |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Field of view of thermal camera and colour camera |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Finding the calibration area |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Objective Statements** | **Task** | WK 18 11/1 | WK 19 18/1 | WK 20 25/1 | WK 21 1/2 | WK 22 8/2 | WK 23 15/2 | WK 24 22/2 | WK 25 1/3 | WK26 8/3 | WK27 15/3 | WK28 22/3 | WK29 29/3 | WK30 5/4 | WK 31 12/4 | WK 32 19/4 | WK 33 26/4 | WK 34 3/5 |
| **Track and display temperature of people at different distances** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Heat sensitive paint colour calibration |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Calibration algorithm for thermal camera |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Debugging and testing of calibration algorithm |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Face tracking |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Displaying final results** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Temperature from thermal camera |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Showing temperature and bounding box in display |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Table (4): Final Project Schedule

# Appendix B – Budget

|  |  |
| --- | --- |
| **Items\*** | Cost |
| **Raspberry Pi 4** | HKD 688 |
| **Heat sensitive (thermochromic) paint** | HKD 326.9 |
| **DS18b20** | HKD 38 |
| **Thermal camera** | RMB 383 |
| **Colour camera** | HKD 244 |
| **Hand warmer** | RMB 30 |
| **TOTAL** | RMB 413  HKD 1679.9 |

Table (5): Budget

# Appendix C – Meeting Minutes

First Meeting

Date: 6/9/2022

Time: 17:00 – 18:00

Location: Room 2451

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* Professor introduced the general direction of the whole project
* Professor hardware idea: thermal cam, colour cam, heat sensitive paint
* Professor suggested alternate approaches and compared them with projects from previous years

|  |  |  |
| --- | --- | --- |
| **Action Item to be completed** | **By when** | **By whom** |
| Finish the proposal report | Sep. 14 th | LI Hoi Him,  LI Hong Yin,  LO Yat Hei |

Second Meeting

Date: 13/9/2022

Time: 17:00 – 18:00

Location: Room 2451

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* All three group members have completed their respective parts of the draft proposal report
* Professor points out minor details to refine in the draft proposal report
* LI Hong Yin suggested using a Peltier element as a constant heat source for calibration and we intend to carry out this method if the performance of the paint is unsatisfactory.

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Finish the proposal report | Sep. 14th | LI Hoi Him,  LI Hong Yin,  LO Yat Hei | In progress  (Draft completed) |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Finish the proposal report | Sep. 14th | LI Hoi Him,  LI Hong Yin,  LO Yat Hei |
| Test colour changing properties of the heat sensitive paint | Sept. 21st | LI Hoi Him |
| Map the colour changes of heat sensitive paint to different temperatures | Oct. 5th | LO Yat Hei |

Third Meeting

Date: 10/10/2022

Time: 13:00 – 18:00

Location: Room 3125

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* LI Hoi Him has completed testing the colour changing properties heat sensitive paint
* LO Yat Hei has completed mapping the colour changes of heat sensitive paint to different temperatures

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Finish the proposal report | Sep. 14th | LI Hoi Him,  LI Hong Yin,  LO Yat Hei | Completed |
| Test colour changing properties of the heat sensitive paint | Sept. 21st | LI Hoi Him | Completed |
| Map the colour changes of heat sensitive paint to different temperatures | Oct. 5th | LO Yat Hei | Completed |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Testing different electric heaters or hand warmers | Oct. 19th | LI Hong Yin |
| Creating an enclosure for the calibration area | Oct. 26st | LO Yat Hei, LI Hoi Him |

Fourth Meeting

Date: 25/10/2022

Time: 17:00 – 18:00

Location: Room 2451

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* Reported findings to professor after testing hardware, which meets current specifications to continue the project
* Professor addressed concerns about the type of thermal camera and software used in the next stage

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Test different electric heaters or hand warmers | Oct. 19th | LI Hong Yin | Completed |
| Creating an enclosure for the calibration area | Oct. 26th | LO Yat Hei, LI Hoi Him | In progress |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Creating an enclosure for the calibration area | Oct. 26th | LO Yat Hei, LI Hoi Him |
| Testing thermal and colour camera | Nov. 16th | LO Yat Hei |

Fifth Meeting

Date: 14/11/2022

Time: 13:00 – 18:00

Location: Room 3125

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* LO Yat Hei and LI Hoi Him have completed creating and enclosure for the calibration area
* LO Yat Hei is making progress testing the thermal and colour cameras
* It is suggested that a hardware upgrade is in order

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Creating an enclosure for the calibration area | Oct. 26th | LO Yat Hei, LI Hoi Him | Completed |
| Testing thermal and colour camera | Nov. 16th | LO Yat Hei | In progress |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Testing thermal and colour camera | Nov. 16th | LO Yat Hei |
| Field of view of thermal camera and colour camera | Dec. 7th | LI Hoi Him |

Sixth Meeting

Date: 16/11/2022

Time: 17:00 – 18:00

Location: Room 2451

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* Professor advised further testing is required as proof of concept before spending a significant part of the budget on a camera with higher resolution
* LI Hoi Him proposed using a square shaped calibration area to track it.
* LO Yat Hei has completed testing thermal and colour camera
* LI Hoi Him is making good progress on field of view of thermal camera and colour camera

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Testing thermal and colour camera | Nov. 16th | LO Yat Hei | Completed |
| Field of view of thermal camera and colour camera | Dec. 7th | LI Hoi Him | In progress |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Field of view of thermal camera and colour camera | Dec. 7th | LI Hoi Him |
| Finding the calibration area | Jan. 4th | LI Hong Yin,  LO Yat Hei |

Seventh Meeting

Date: 22/12/2022

Time: 13:00 – 16:00

Location: Room 3125

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* LI Hoi Him has completed cropping the field of view of thermal camera and colour camera.
* LI Hong Yin and LO Yat Hei are making good progress on finding the calibration area. There are still some small discrepancies based on the position difference of the cameras and the sensitivity of the software.
* LI Hoi Him will perform heat sensitive paint calibration after a different paint more in line with our current requirements is obtained.

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Field of view of thermal camera and colour camera | Dec. 7th | LI Hoi Him | Completed |
| Finding the calibration area | Jan. 4th | LI Hong Yin,  LO Yat Hei | In progress |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Finding the calibration area | Jan. 4th | LI Hong Yin,  LO Yat Hei |
| Heat sensitive paint calibration | Feb. 1st | LI Hoi Him |

Eighth Meeting

Date: 15/2/2023

Time: 14:00 – 17:30

Location: Room 3125

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* LI Hong Yin and LO Yat Hei have completed the software that allows the camera to locate the calibration area.
* LI Hoi Him has successfully assigned the appropriate RGB values to the heat sensitive paint, though there are still some environmental factors such as lighting that would affect it.
* LI Hong Yin will create the calibration algorithm.
* LI Hoi Him will test and debug the algorithm to reduce error and improve accuracy.

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Finding the calibration area | Jan. 4th | LI Hong Yin,  LO Yat Hei | Completed |
| Heat sensitive paint calibration | Feb. 1th | LI Hoi Him | Completed |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Calibration algorithm for thermal camera | Mar. 8th | LI Hong Yin |
| Debugging and testing of calibration algorithm | Apr. 5th | LI Hoi Him |

Ninth Meeting

Date: 15/3/2023

Time: 14:00 – 17:30

Location: Room 3125

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* LI Hong Yin has completed the calibration algorithm for the thermal camera.
* LI Hoi Him is making progress with testing and debugging with the calibration algorithm, which is currently not accurate enough are requires improvement.

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Calibration algorithm for thermal camera | Mar. 8th | LI Hong Yin | Completed |
| Debugging and testing of calibration algorithm | Apr. 5th | LI Hoi Him | In progress |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Debugging and testing of calibration algorithm | Apr. 5th | LI Hoi Him |

Tenth Meeting

Date: 1/4/2023

Time: 13:30 – 18:00

Location: Room 3125

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* LI Hoi Him is making good progress with testing and debugging with the calibration algorithm, which should be complete by the next meeting
* LI Hoi Him will

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Debugging and testing of calibration algorithm | Apr. 5th | LI Hoi Him | In progress |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Debugging and testing of calibration algorithm | Apr. 5th | LI Hoi Him |

Eleventh Meeting

Date: 5/4/2023

Time: 13:30 – 18:00

Location: Room 3125

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei

Absent: N/A

Minutes taken by: LO Yat Hei

* LI Hoi Him will is making progress with testing and debugging with the calibration algorithm, which is currently not accurate enough are requires improvement.

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Debugging and testing of calibration algorithm | Apr. 5th | LI Hoi Him | In progress |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Debugging and testing of calibration algorithm | Apr. 5th | LI Hoi Him |
| Face tracking | Apr. 19th | LI Hoi Him |

Twelfth Meeting

Date: 13/4/2023

Time: 14:00 – 16:00

Location: Room 2451

Attendees: LI Hoi Him, LI Hong Yin, LO Yat Hei,

Absent: N/A

Minutes taken by: LO Yat Hei

* Reported project progress to professor.
* Professor suggests testing on assumptions such as the exact temperature of change for HSP and the rate of change.
* Professor suggests comparison with existing calibration methods to prove that our approach is an improvement.
* Professor points out other potential applications besides measuring human body temperature, such as cooling systems for electronic equipment.

Action Items from Previous Meeting

|  |  |  |  |
| --- | --- | --- | --- |
| Action Item to be completed | By when | By whom | Status |
| Debugging and testing of calibration algorithm | Apr. 5th | LI Hoi Him | Completed |
| Face tracking | Apr. 19th | LI Hoi Him | Completed |

Action Items for Next Meeting

|  |  |  |
| --- | --- | --- |
| Action Item to be completed | By when | By whom |
| Finish final report | Apr. 19th | LI Hoi Him, LI Hong Yin, LO Yat Hei |
| Test and refine results | May. 4th | LI Hoi Him, LI Hong Yin, LO Yat Hei |

Appendix D– Group Members’ Contribution **Li Hong Yin**

I am responsible for the software part of the project, and we have made great progress in terms of the setup of the MLX90641 thermal camera, setup of communication between Raspberry Pi and laptop, and the integration and display of both cameras, using Python as our main programming language due to the availability of existing libraries, our previous programming experience, and not having to build the entire directory every time we make changes on our Raspberry Pi.

**Thermal Camera**

The MLX90641 thermal camera is connected and set up with our Raspberry Pi 3B with the provided library. First, we confirmed that the camera works and can give an output by following an online tutorial. Then, I modified the code so that the camera would output the temperature data and send it to the computer to be used for calibrations later. Since the Raspberry Pi 3B does not have a lot of memory, I got rid of some unnecessary code, and streamlined it to send the full frame data to the computer at a maximum of 4 frames per second.

**RGB Camera**

For the RGB camera, we decided against purchasing a new camera and instead used our laptop cameras and an existing Logitech Webcam we own. By simply using OpenCV’s video capture, we have determined that the Logitech Webcam is not good for preliminary testing, since it would take more than a minute to start up the capture window. After deciding on the RGB camera, we would move on to getting the thermal camera output to the laptop. For the final display, we moved to the Logitech webcam for a more accessible and portable camera.

**Communication between devices**

To retrieve the thermal camera output to be used on the laptop, I have set up the laptop to host a Wi-Fi hotspot which the Raspberry Pi is able to join. Using the Python ‘socket’ library, I set up a connection between the laptop and the Raspberry Pi, with the laptop being the server (host). The laptop would receive data until the entire pickled 768-length array containing all the data points from the thermal camera is received. The data is then reshaped into its original 24x36 array, and made into a gray-scaled image using a linear intensity level normalized to the temperature data. The image is then scaled up 20 times and cropped to make it visible have the same resolution as the Logitech webcam. With this, we can have a live feed of the thermal camera captured image on our laptop and allow us to do any processing necessary on the relatively more powerful computer.

**Integration**

After getting the code from my teammates for face recognition and calibration plane detection, I put them into the same loop to create a live feed of both the thermal and RGB camera, and all the extra processing. The computer could display following: 1) Image from RGB camera. 2) Image from thermal camera. 3) Bounding boxes around detected faces and calibration plane on image from RGB camera. After setting up the thermal camera and webcam, cropping the images respectively, and matching the frames, we have a rough mapping between the two cameras effective within 3 meters. With that, we can transfer the coordinates from the RGB image to the thermal image, and locate and draw bounding boxes around faces on the thermal camera image.

**Calibration and display**

The calibration algorithm first detects the calibration frame, with a failsafe and can be manually reset if needed, and then compares the states of the paints to the last frame. Any differences would result in recalculating the temperature delta as the offset. The delta is shown next to the calibration area and the temperatures extracted from the faces are calibrated with the offset and shown next to the person tracked.

**Li Hoi Him**

I am responsible for the image processing algorithms of the project.  The tracking programs were able to provide valuable information for our purposes by highlighting targeted objects for temperature measurement using a simple camera, as well as processing the image data into processable numerical data.  The programs were written with the help of libraries such as OpenCV in Python to be compatible with the Raspberry Pi board that controls the thermal camera and other programs written by my teammates.

**Video Capture**

A live video feed is required for any image processing to take place. That being said this part of the program is rather straight forward. I captured the frame from the laptop camera once per loop and greyscaled it for easier processing. The data collected by the tracking algorithms, i.e. the sizes and coordinates of identified faces, is then directly mirrored back to the coloured image as an output.

**Face Tracking**

Face tracking is needed to identify humans from other objects measure an individual’s body temperature. The program compares enclosed shapes in the image with the massive OpenCV library to check if the said shape is a face or not.

**Calibration Area Tracking**

Other than humans, the thermal camera also needs to locate the calibration area such that it can pay attention to the colour changes in the heat sensitive paint for thermal calibration to take place.  Just like the face tracking program, the program tries to search for a specific shape to locate the targeted object.  Instead of shapes of faces, I have designed a custom “QR pattern like” shape to be identified by the program in a noisy environment.

I programmed the machine to look for a shape that meets the following requirements, i) the shape is a square with exact 1:1 dimension, ii) the shape is in a specified location, i.e. its coordinates are in a specific range, iii) there are other squares that meet the first requirement within this shape, and iv) all these shapes have their centre points at the same coordinates.  In order to assist the program in finding this QR pattern, the inner and smaller square had a thicker border, such that the exterior outline and interior border are seen as 2 separate squares by the camera.  With the pattern located, we can estimate the size and location of the calibration area, given the assumption of 1) the calibration is set upright, 2) the QR pattern is located on the top left corner of the calibration area, 3) the calibration pattern is also a square, and 4) the width and height of the calibration area is exactly 4 times longer than that of the QR pattern.  The program finds the top left corner of the QR pattern and bounds the estimated calibration area with a square 16 times larger than the QR pattern in area.  We would then be able to read light values from this bounded area to measure the temperature of the heat sensitive paint based on its colour.

**Heat Sensitive Paint Calibration**

The thermal camera is calibrated based on the visual data provided by the colour of the heat sensitive paint. To convert the RGB values of the 3 painted patterns into processable temperature data, I first extracted one of the RGB values of each paint respectively. By only storing one value instead of all 3, we could utilize data more effectively by reducing the amount of excess data and the program could run more smoothly. For the green paint only the G value was stored, the R value for the red paint, and lastly the B value for the blue paint. Then, by observing which paint has it value increase beyond a threshold integer, we deduce that said paint has changed into a lighter colour- a reaction from the paint reaching a set temperature. Based on which paint has changed colour, we would deduce what temperature the calibrator has reached. If only the green paint has changed colour, the calibrator has reached 35oC, which is the colour-changing point of the green paint. If the blue paint starts turning lighter, we will deduce the calibrator has reached 38oC based on the same logic. However, given that the heat source we use cannot reach 45oC, we would deduce something has gone wrong or some external variables has affected the calibrator, i.e. a powerful external heat source/light source, despite the readings of the other 2 paints. Hence, the red paint acts as a debug tool rather than a temperature indicator.

**Lo Yat Hei**

I am responsible for the hardware part of the project, namely testing and applying the heat sensitive paint and heat source.

**Heat Sensitive Paint**

As the core of the project, it is important to ensure that the heat sensitive meets our specifications to carry out the calibration of the thermal camera. Thus, I performed various tests on the paint to understand their properties and determine if they were viable for use in the project. Three different types of paint were selected to be used in our final design.

The colour of the first type of paint has a sharp changing point at around 35°C as advertised, changing from dark blue to pale yellow when heated. The colour change is quick and reversible. These colour changing properties are not affected by the thickness or wetness of the paint and persist for months after it is applied.

The paint is viscous and insoluble in water and alcohol, and it adheres well to the aluminum foil I applied it on. These properties mean that the paint cannot be easily removed and is suitable for creating the calibration area.

The second type of paint shared similar properties with the first, but it changes colour at 38°C from dark blue to pale blue. The third type of paint changes at 45°C from dark brown to red, is soluble in water and is less viscous compared to the other two.

I applied the dark blue-to-pale yellow and dark blue-to-light blue paint with a cotton swab instead of a paintbrush because it is difficult to clean it afterwards. I applied a thick layer of paint and made sure it is distributed evenly so not affect the reception of the thermal camera and prevent reflections from the aluminum foil.

**Heat Source**

While a blackbody heat source is unavailable due to its high cost, which is why heat sensitive paint is used to show the temperature of the heat source and calibrate the thermal camera, it is desirable to use a heat source which can provide a stable heat output close to human body temperature.

Initially, an electric hand warmer was proposed to be used as the heat source, as it used for similar projects from previous years and can maintain a range of temperatures close to human body temperature above 35°C. Unfortunately, we did not have any electric heaters at our disposal, and it was difficult to purchase one at the beginning of the project before winter.

Therefore, iron powder heat packs that I purchased last year were used as the heat source of the project. The packaging of the iron powder heat packs states that it can go up to 67°C and maintain an average of 53°C, but according to the experiments I performed, the actual temperature of the iron powder heat packs was lower, fluctuating between around 40°C and 60°C, possibly due to oxidization of the iron powder. This temperature range was acceptable at that stage of the project, which was testing the thermal camera and heat sensitive paint, the iron powder heat packs provided a heat source above human body temperature and proved the capabilities of the heat sensitive paint and thermal camera.

Compared to electric heaters, iron powder heat packs cost less and do not require a charging and an extra energy source, which would complicate the calibration setup. On the other hand, the temperature output of iron powder heat packs is less stable, and heat dissipates more quickly.

A hairdryer was also used at a heat source during testing. It heats up much faster than the heat packs and can reach higher temperatures, so it was useful for testing the properties of the paint and the software. However, the temperature output was unstable, it was too hot compared to the human body, and the wind made it unideal for calibration purposes.

# Appendix E – Deviations from the proposal and progress reports and supporting reasons

There are several changes made to the project compared to our original plans from the proposal and progress reports. One notable deviation is creating a palette of three different colours with three different kinds of heat sensitive paint, each changing colour at a temperature different from the others, instead of using only one. This change was made to improve the accuracy of our calibration system, as the paint from our original plan changes colour at 35°C, which is not very useful for reference when determining whether someone has a fever (37.5°C). Aside from getting more accurate results, the new system can also solve some issues such as lighting and reflections, as the heat pack is normally insufficient to change the colour of the 45°C paint, so if the camera captures a colour change of it, it may be due to lighting issues and the result would be ignored.

Furthermore, we have decided to use the Logitech webcam for our final setup, since it would provide easy positional adjustment, and it would be easier to attach the thermal camera to it. It gives us a more reliable and accurate result for matching up the frames of the colour camera and thermal camera.

There are also some minor changes in the schedule and distribution of work, due to some tasks being completed earlier than expected and others taking more time than planned.