UNIVERSIDADE D
COIMBRA

Ana Rita Mendes Barradas

DEVELOPMENT OF A NON-INVASIVE TEMPERATURE MONITORING AND TRACKING DEVICE USING MACHINE LEARNING ALGORITHMS

Thesis submitted to the University of Coimbra in fulfillment of the requirements for the Master's Degree in Engineering Physics under the scientific supervision of Prof. Filipe Manuel Almeida Veloso and Prof. Francisco Filipe Bento Neves

September of 2022

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

Development of a non-invasive temperature monitoring and tracking device using machine learning algorithms

Supervisor:

Prof. Filipe Manuel Almeida Veloso

Prof. Francisco Filipe Bento Neves

Jury:

Prof. António Adriano Castanhola Batista

Prof. Custódio Francisco Melo Loureiro

Prof. Filipe Manuel Almeida Veloso

Dr. Miguel Crispim Romão

Dissertation submitted in partial fulfillment for the degree of Master of Science in Engineering Physics.

Coimbra, September 2022

Acknowledgments

A realização desta tese não seria possível sem o apoio, compreensão, e persistência dos meus orientadores Filipe Veloso e Francisco Neves. A eles, o meu profundo agradecimento.

Ao Mário, à Joana e ao Zé, por contribuirem para a validação dos resultados da minha dissertação.

Agradeço à minha família,: mãe, pai, tia, Ei, avó Alice, Sara e Pedro. Sem o vosso esforço, apoio e dedicação incondicionais, nada do que alcancei seria possível. Serão sempre a minha maior força.

Agradeço à Tarina, a minha irmã, que foi, é e será o meu maior exemplo para sempre. Nenhum passo do meu crescimento ou conquista fariam sentido sem serem partilhados contigo, ou vividos sem ti.

Agradeço em especial à Maria, ao Hugo e à Seo, por terem sido os melhores amigos e companheiros de todas as horas ao longo destes anos, desde o primeiro momento. Ao Joca e ao Zé, por me acompanharem em fases importantes ao longo do meu percurso, por me demonstrarem apoio, e por terem sido companheiros incríveis, dos quais levo excelentes momentos. Ao Gonçalo, por ter sido a minha maior fonte de crescimento. Gratidão vai ser sempre uma palavra que parece pouco.

A todos os que fizeram parte da minha passagem pelo mundo do associativismo (NEDF, Physis e Pollux) e que, de alguma forma, contribuiram para o meu desenvolvimento pessoal e profissional.

Resumo

A medição de temperatura de forma não invasiva tem muitas aplicações, desde a biologia (e.g. monitorização de aves em nidificação) à indústria (e.g. produtos em tapetes rolantes) ou medicina (e.g. medição de febre).

O desenvolvimento de bibliotecas para visão por computador ou *machine learning* em *open source*, como o OpenCV permitem o rápido processamento por meio de câmeras e computadores simples, como é o caso do RaspberryPi. Combinando esta capacidade com os dados adquiridos através de uma câmera de infravermelho (IR) torna-se possível monitorizar a temperatura de objetos ou corpos.

Neste projeto, foi desenvolvido um sistema de monitorização de temperatura em tempo real, orientado para o reconhecimento facial de individuos e a medição da temperatura das suas testas. O sistema é composto por um RaspberryPi e a sua câmera (*Module 2*) e uma câmera térmica MLX90640.

Numa fase inicial, foi estudado o comportamento da camera térmica, a partir de modelos existentes para o seu funcionamento. Foram analizados os efeitos da distância na aquisição da temperatura, através de seleções manuais da região de interesse (RoI) em imagens térmicas, onde foi avaliada a incerteza resultante da escolha da seleção do RoI e a incerteza estatística, avaliando a evolução destas incertezas com o aumento da distância. Através do conhecimento da influência do ambiente na medição de temperatura por imagens infravermelho, foi testado um modelo existente para compensar estes efeitos com a distância, onde se verificou uma melhoria relativamente ao erro absoluto médio calculado. O algoritmo de deteção de faces usado foi o classificador *Haar-Cascade*, implementado no OpenCV. Para teste final do sistema, foram objeto de estudo um ambiente simulado e outro com faces reais (com uma taxa de amostragem de 32Hz e 8Hz, respetivamente). O sistema foi, por fim, validado com indivíduos escolhidos aleatoriamente para o efeito. As características do sistema e as suas limitações são finalmente sumarizadas a partir dos resultados obtidos.

Abstract

Non-invasive temperature measurement has many applications, from biology (e.g. monitoring of nesting birds) to industry (e.g. products in a conveyor belt) or medicine (e.g. fever checking).

The development of open-source computer vision (CV) and machine learning (ML) libraries, as OpenCV, allow the fast process of images using simple cameras and computers, such as RaspberryPi. Combining these capabilities with the data acquired through an infrared (IR) thermal camera it is possible to monitor objects and bodies.

In this master's thesis a real time temperature-monitoring system was developed, oriented towards facial recognition and to measure the subject's forehead temperature. The system is composed by a RaspberryPi and its camera (*Module 2*) and a thermal camera MLX90640.

As a first step, the characteristics of the thermal camera were studied, from existing models describing its performance. The effects of the distance on temperature acquisition were studied, using manual selections of regions of interest (RoI) from thermal images, where the uncertainty on the selection of the RoI and the statistical uncertainty were evaluated, studying the evolution of these uncertainties as the distance increases. Existing models studying the influence of the environment on the temperature measurements through IR imaging have been tested, in order to estimate the influence of the subject-camera distance. The used algorithm for face detection was the *Haar-Cascade*, implemented at OpenCV. For the final test of the system, a study was made under simulated conditions, using both simulated and real faces (with a refresh rate of 32Hz and 8Hz, respectively). Finally, the system was validated with randomly picked subjects. The system's characteristics and its limitations were then sumarized from the obtaind results.

"You've had a landscaper and a house keeper since you were born

The starshine always kept you warm

So why see the world, when you got the beach?"

Frank Ocean

Contents

List of Acronyms

CV	Computer Vision
FPS	Frames per second
IR	Infrared
ML	Machine Learning
RMS	Root mean square
RoI	Region of interest

List of Figures

List of Tables

1 Introduction

1.1 Context and motivation

Long-distance temperature measurement techniques are quite useful today, especially if there is the need to avoid contact among instruments and persons or objects.

Several applications can be made by combining cameras and their potentialities. In terms of non-invasive temperature monitoring and tracking through computer vision (CV) techniques, significant advances have been made mainly due to the pandemic context and other fast spreading diseases [1]. These techniques are important to monitor the body temperature of moving bodies, for instance, in an airport gate. Other examples of medical applications are monitoring of temperature variations on the skin and other body components, complementing the existing imaging techniques. This can lead to the detection of diseases at an early stage [2]. Surprisingly, these techniques are also used in the food and agriculture industry, for example, for the detection of spreadly fungi [2]. In other industry applications, thermal imaging techniques are used to detect hot spots in electrical systems, that indicate a possible system's breakdown [2]. Other applications are related to civil engineering, particularly in inspections to study the heat losses on buildings [2]. It may also be used in fire detection [3].

The images acquired by an optical camera in the RGB (Red, Green, Blue) space, usually carry a lot of information concerning objects colors, textures and shapes. In contrast, the information from thermal cameras is usually represented using a grayscale image[1] allowing only to distinguish objects at different temperatures. Although textures, the environment's illumination conditions, and contrasting colors are good features for object detection, they can also make this job harder. By combining thermal and optical images, and since thermal images avoid the mentioned problems, we can improve our results of object detection and temperature measurement [4].

Considering the Stefan-Boltzmann law, which tells us that every object or body emits thermal radiation proportional to its temperature, we can perform temperature measurements and analysis

[1] Thermal images are often represented with a color scale, that allows an easier way for interpretation

from thermal images.

Nevertheless, thermal noise from surrounding objects and bodies[2], contributing to the global emittance of the environment, must be taken into account as the distance between the object and the camera increases. To account for those effects a temperature model [1], must be considered.

In order to have a highly precise temperature measurement, high-resolution cameras need to be used. The higher the resolution and precision, the more expensive cameras are. In the case of thermal cameras, the ones with higher resolution are still very expensive. So, there is a big challenge when trying to obtain similar good results when using a thermal camera with a lower resolution in a low-cost system.

Since different cameras have different intrinsic properties and technical characteristics, there are some challenges when combining thermal and optical images. For that purpose several matching [5] and calibration methods [6] are available in the literature.

In this work we will:

- combine the acquired images from MLX90640 and RaspberryPi low-cost and low-resolution IR and optical cameras, respectively;
- propose a homography-based calibration method;
- evaluate and discuss the system's performance when applying object detection algorithms.

1.2 Analysis requirements

The system, based on a RaspberryPi, must:

- be capable to identify persons or objects and measure and report the their temperature in their natural environment with a precision <1ºC;
- have an analysis rate >1 Hz;
- be built by asserting on possibility of scaling up with bigger computers if required by the application.

[2]Thermal noise may be from surrounding objects and bodies, which contributes to the global emittance on the environment.

2 Theoretical introduction

2.1 Pinhole camera model

Before introducing the mechanisms used for temperature measurement, a camera model must be considered, in order to understand the principles beyond the camera's image acquisition. For camera calibration, a *pinhole camera model* is considered, the simplest model of representing a camera.

This model considers an **object** (X), a **pinhole plane**, and an **image plane**, as shown in figure 2.1. Considering the pinhole plane, there is a very tiny aperture that allows light to go through it. The object's image is then projected in an image plane, also called a **projective plane**.

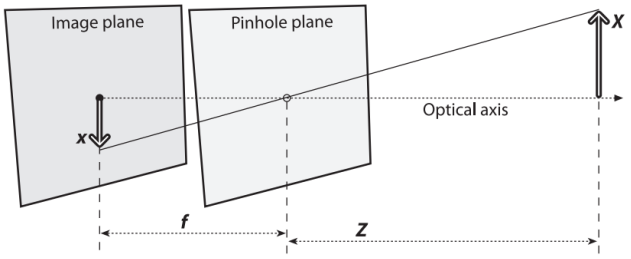


Figure 2.1: Pinhole camera model, taken from [7].

The relation between the parameters mentioned in figure 2.1 is expressed by:

$$x = -f\frac{X}{Z} \tag{2.1}$$

where f is the focal length, X is the length of the object, x is the object's image length and Z is the distance between the camera and the object.

By repositioning the pinhole camera's configuration, as shown in figure 2.2, the mathematical and geometrical interpretation gets simpler. Note that now image and pinhole plane are swapped. In this case, the aperture on the pinhole plane is called the ***center of projection***. The optical axis and the image plane intersect at the ***principal point***, q. In this configuration, in the new image plane, the size of the object's image is the same as the object. As a consequence, the relation is now $\frac{x}{f} = \frac{X}{Z}$.

Considering a point Q with coordinates (X,Y,Z), its projected image on the projection screen[1] is located at $(x_{\text{screen}}, y_{\text{screen}})$. Following the equation 2.1, and by introducing new parameters, we establish:

$$x_{\text{screen}} = f_x \frac{X}{Z} + c_x \tag{2.2}$$

$$y_{\text{screen}} = f_y \frac{X}{Z} + c_y \tag{2.3}$$

The (x_{screen}, y_{screen}) differs from the actual (x, y) shown in the figure 2.2. In the screen configuration the points c_x and c_y, defined as the image center, also do not coincide with the principal point, since the center of the camera's sensor is usually not on the optical axis. However, to simplify this model, we'll consider by approximation that the *principal point* coincides with the center of the sensor [7].

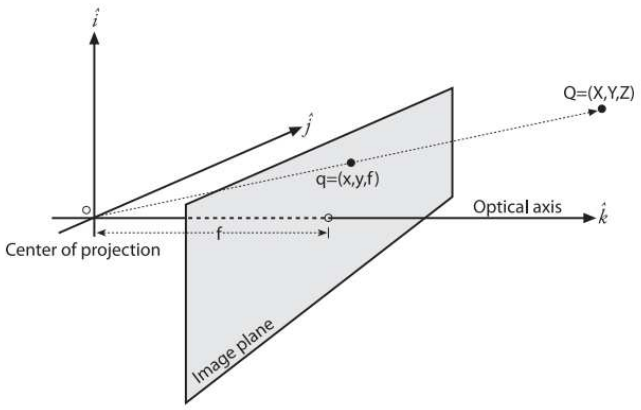


Figure 2.2: Alternative configuration of the pinhole camera model, taken from [7].

Now the relation is positive, as shown in figure 2.2.

From equations 2.2 and 2.3, we must note that two different focal lengths were introduced. They were introduced due to low-cost camera specifications[2], that have different focal lengths [7]. These relations are called *projective transforms*. When working with these projective transforms, *homogeneous coordinates*[3] are convenient to use [7], since they enable the implementation of common operations such as translation, rotation, scaling and perspective projection, as a matrix operations.

[1] The image plane is the projection screen "pushed" in front of the pinhole [7]

[2] "The individual pixels on a typical low-cost imager are rectangular instead of square" [7].

[3] When working with 3D coordinates, Euclidian geometry is commonly used. In *projective geometry.*, often used on computer vision applications, there is an extra dimension, called *projective space*, and its coordinates are called **homogeneous coordinates** (as Cartesian coordinates are used in Euclidean geometry)

When using homogeneous coordinates the camera is defined by a 3×3 matrix, which is called the ***camera intrinsic matrix***. This matrix translates how the points in the physical world are projected into the camera:

$$q = MQ \tag{2.4}$$

where

$$q = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \; M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \text{ and } Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.5}$$

2.2 Distortion

Despite its theoretical advantages, the model presented above does not work very well in the real conditions and lens are required to gather more light.

However, due to their physical properties, lenses introduce radial and tangential distortions. Also lens are not often centered and parallel to the image plane. This introduces tangential distortions (see figure 2.3). The shape of the lens brings radial distortions (see figure 2.4).

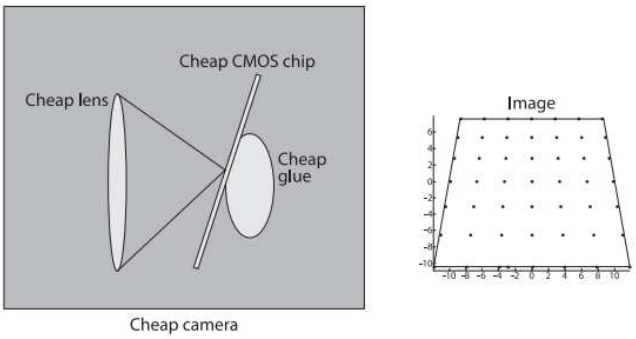


Figure 2.3: Example of tangential distortion, taken from [7].

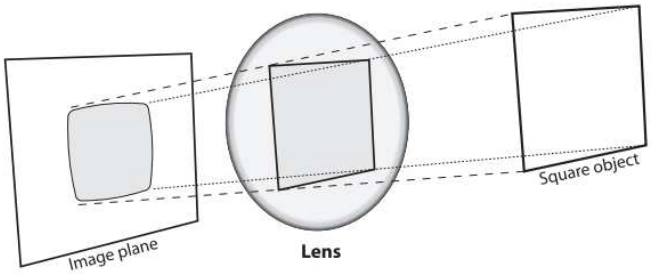


Figure 2.4: Example of radial distortion, taken from [7].

For radial distortions, the distortion increases as we move toward the periphery, being considered 0 at the optical center of the image. Having this in mind, and considering that the distortion is small, it can be characterized by a Taylor series expansion around the distortion center.

The Brown-Conrady model [8, 9] characterizes radial and tangential distortions as a Taylor series. From Taylor series, a relation between the coordinates of the distorted image (x, y) and the undistorted coordinates (x_r, y_r) is established:

$$x_r = x(1 + k_1 r^2 + k_2 r^3 + k_3 r^4) \tag{2.6}$$

$$y_r = y(1 + k_1 r^2 + k_2 r^3 + k_3 r^4) \tag{2.7}$$

In a similar way, for tangential distortion, the following relations can be established:

$$x_t = x + [2p_1 y + p_2(r^2 + 2x^2)] \tag{2.8}$$

$$y_t = y + [p_1(r^2 + 2y^2) + 2p_2 x] \tag{2.9}$$

The k_i and p_i parameters are the radial distortion and tangential distortion coefficients, respectively. In both equations, r is defined as on equation 2.10, where x_c and y_c are the coordinates of the distortion center:

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{2.10}$$

As can be seen from the previous equation, five parameters are used to define the camera's intrinsic parameters. From these parameters, a 5×1 a *distortion vector* is defined as $[k_1\ k_2\ p_1\ p_2\ k_3]^T$.

2.3 Camera calibration

In order to perform the calibration of the cameras, the intrinsic parameters of the cameras and the distortion vector must be obtained. OpenCV's library supplies calibration algorithms that provide a way of determining these parameters and vectors. In these algorithms, different perspectives of an object must be considered in order to identify as many strategic points as possible. Translation and rotation processes are very important procedures for the calibration step.

2.3.1 Rotation matrix

Rotation is an important step for the camera calibration. By applying rotation matrices to a point's location, a new description of it is obtained, but in a different coordinate system. Considering a rotation around x, y, and z axes with rotation angles ψ, φ, and θ, a rotation R is described by the product of the matrices presented in equations 2.11, 2.12, and 2.13, *i.e.* $R = R_x(\psi)R_y(\varphi)R_z(\theta)$:

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\psi & sin\psi \\ 0 & -sin\psi & cos\psi \end{bmatrix}, \tag{2.11}$$

$$R_y(\varphi) = \begin{bmatrix} cos\varphi & 0 & -sin\varphi \\ 0 & 1 & 0 \\ sin\varphi & 0 & cos\varphi \end{bmatrix}, \text{ and} \tag{2.12}$$

$$R_z(\theta) = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.13}$$

Figure 2.5 illustrates the rotation for the Z axis, which can be then related to other coordinate changes (for X and Y axis).

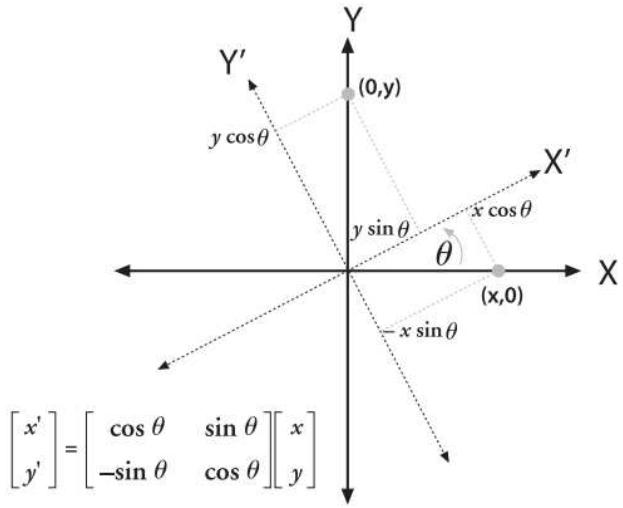


Figure 2.5: Coordinate change by applying rotation to the Z axis, taken from [7].

2.3.2 Translation

The translation vector represents a change from one coordinate system to another with the origin shifted to a different point. Considering two different coordinate systems, one centered in

the object and the other on the camera, the translation vector is represented by the equation 2.14

$$T = o_{\text{object}} - o_{\text{camera}} \tag{2.14}$$

where o_{object} and o_{camera} represent the origin of the each correspondent coordinate system.

The transformation between two coordinate systems is completed by combining rotation and translation. Having an object-centered and a camera-centered coordinate systems, the relation between two points is represented by expression 2.15. Figure 2.6 shows a representation of the mentioned transformation.

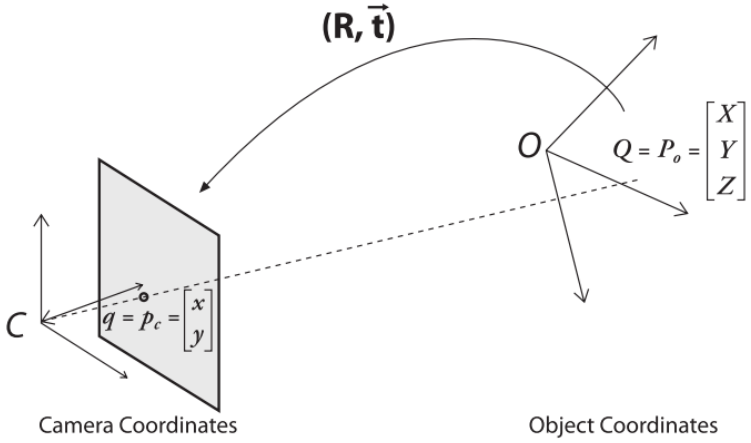


Figure 2.6: Conversion to camera coordinate system, by applying rotation and translation operations, taken from [7].

$$P_{\text{camera}} = R(P_{\text{object}} - T) \tag{2.15}$$

2.4 Homography

In CV, planar homography is defined as an invertible transformation of points between two images on a projective plane. A 3x3 homography matrix specifies the mapping between two images of the same plane as viewed by a pinhole camera. Using homogeneous coordinates [7], a homography relation can be established by matrix multiplication, considering an object plane and the camera plane:

$$\tilde{q} = H\tilde{Q} \tag{2.16}$$

where H is the homography matrix, $\tilde{q}$ is a point on the camera's plane, and $\tilde{Q}$ is the corresponding point on the object's plane, and are represented by:

$$\tilde{q} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ and } \tilde{Q} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{2.17}$$

The homography matrix can be decomposed into a multiplication of two different matrices (W and M). The first one (W) contains the physical transformation of the points under analysis. It combines the *rotation* and *translation* relations in one matrix, since we are dealing with homogeneous coordinates.

$$W = \begin{bmatrix} R & T \end{bmatrix}, \tag{2.18}$$

where R and T are the rotation matrix the translation vector, respectively. The W matrix, in the equation 2.18 is the **extrinsic matrix**. This matrix has to do with the camera's *intrinsic parameters*, already mentioned in section 2.1 and in Equation 2.5, and it is represented as:

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.19}$$

Thus, to fully calculate the transformation between $\tilde{q}$ and $\tilde{Q}$, the intrinsic matrix M (2.19) is multiplied by the extrinsic matrix W (2.18), obtaining $\tilde{q} = MW\tilde{Q}$.

Although $\tilde{Q}$ is defined for all space, the interest is in a coordinate representing the plane we are looking at, defined as $\tilde{Q}'$. Considering that the object's plane is defined as $Z = 0$ [7] and taking the rotation matrix as matrix composed by 3×1 columns ($R = [r_1 r_2 r_3]$), the relation 2.20

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = M \begin{bmatrix} r_1 & r_2 & r_3 & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = M \begin{bmatrix} r_1 & r_2 & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \tag{2.20}$$

where H is the **homography matrix**, which describes the transformation between a plane containing the object's points and the image plane. Then, rewriting Equation 2.16 as a function of $\tilde{Q}'$, one gets:

$$\tilde{q} = H\tilde{Q}' \tag{2.21}$$

.

The points of the source object's plane, p_{src}, and the destination image plane, p_{dst}, can be then related by the equation $p_{\text{dst}} = H p_{\text{src}}$. As mentioned, the homography is an invertible transformation, so the previous expression can be rewrited as $p_{\text{src}} = H^{-1} p_{\text{dst}}$. p_{dst} and p_{src} are defined similarly as $\tilde{q}$, on equation 2.17.

OpenCV provides algorithms to obtain these homography matrices, through the function *cv.findHomography()*. As mentioned, the H matrix relates two positions of the points on the source image plane and on the object plane. In CV applications, however there is no explicit need to know the intrinsic parameters. Instead it is usually used to find them using sets of images.

2.4.1 Two-camera system

Epipolar geometry

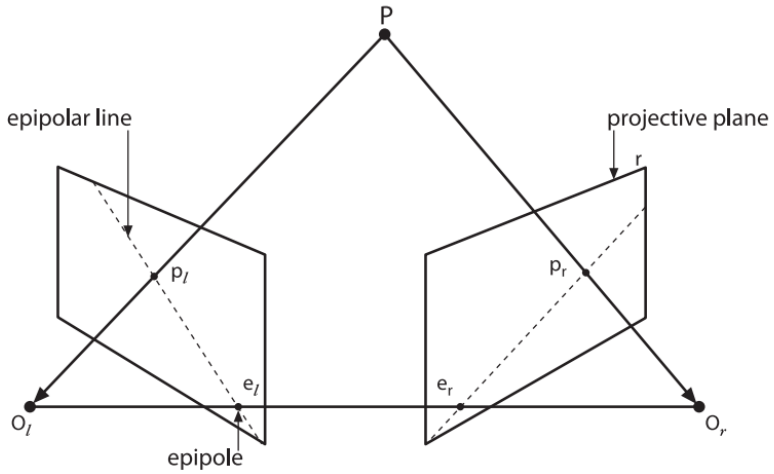


Figure 2.7: Basic elements of epipolar geometry, taken from [7].

Epipolar geometry refers to how an image obtained from one camera corresponds to another image, captured by a second camera. Figure 2.7 shows a simple scheme describing the epipolar geometry principles.

Before introducing these principles, a two-pinhole-camera model (one for each camera) [7] system must be considered. The cameras are represented by their center of projection, O_l and O_r, where l and r represent the left and right cameras, respectively. The projective planes, Π_l and Π_r (see on Figure 2.8), already explained in section 2.1, are also represented on the image. In the scheme (2.7), two cameras see the same point P located at a 3D position in the physical world. This point has a projection on the projection planes, that is represented by the points p_l and p_r, respectively. Furthermore, each center of projection O_l and O_r also has its own projection on the projective planes Π_l and Π_r of the opposite camera, e_l and e_r. These points are the ***epipoles***.

Taking into consideration the three points P, e_l, and e_r, a plane is formed, which is called ***epipolar plane***. On the other hand, by corresponding the points e_l and p_l (e_r and p_r for the right side), we have lines called ***epipolar lines***.

The interest in epipolar geometry relies on finding a relative position of the same point P on different cameras, i.e. being able to compute the transformation of the same point between the projective planes p_l and p_r.

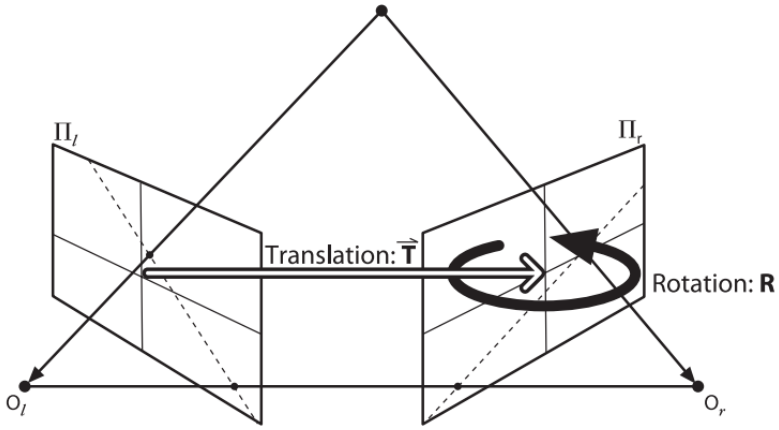


Figure 2.8: Translation and rotation relations on epipolar geometry, taken from [7].

From the epipolar plane, an ***epipolar constraint*** is built. Since we can not deal with 3D coordinates of the physical world, let's consider the physical locations of the point P in the coordinate frames of the cameras: P_l and P_r [4]. The transformation between the points P_l and P_r can be computed by knowing the translation and rotation operation between these two points. Considering the represented point P_r from the right camera, and considering the mentioned operations, a relation about how the point P_r is transformed from P_l is established:

$$P_r = R(P_l - T), \tag{2.22}$$

where R and T represent the rotation and translation operations, respectively.

The epipolar constraint is defined in terms of the dot product of a vector $\mathbf{x}$ that represents any of the points on the epipolar plane with a normal vector $\mathbf{n}$ that passes through a point $\mathbf{a}$ [7]:

$$(\mathbf{x} - \mathbf{a}) \cdot \mathbf{n} = 0 \tag{2.23}$$

Making use of equation 2.22 and 2.23 above, the condition 2.24 can be written, for all possible points P_l passing through the point T, where T is the location of the other camera:

[4] Do not confuse with p_r and p_l that are the projection of the point P on the projective planes.

$$(P_l - T)^T \cdot (T \times P_l) = 0. \tag{2.24}$$

In order to obtain the relation that transforms the image coordinates of one camera to the other, q_l and q_r, the relation between P_l and P_r must be determined. From equation 2.22, and rewriting it as:

$$R^{-1}P_r = (P_l - T), \tag{2.25}$$

and since R is an orthonormal matrix, $R^T = R^{-1}$, and from equation 2.24, one can write:

$$(R^T P_r)^T (T \times P_l) = 0. \tag{2.26}$$

The $(T \times P_l)$ cross product can be written as a product SP_l, where S is a matrix with the following representation.

$$S = \begin{bmatrix} 0 & -T_x & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}. \tag{2.27}$$

Out of expression 2.26, and attending to the algebra property $(AB)^T = B^T A^T$, the follow relation is obtained:

$$P_l^T RSP_l = 0 \tag{2.28}$$

Therefore, the ***essential matrix***, $\boldsymbol{E}$, is defined as the product RS, and the equation 2.28 is equivalent to $P_l^T E P_l = 0$. Despite this relation, we are still lacking an expression that relates p_l and p_r. From the mentioned relation on the beginning of this section, equation 2.1, the relation 2.28 can be rewritten as it follows.

$$E = RS = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 0 & -T_x & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \tag{2.29}$$

$$p_r^T E p_l = 0 \tag{2.30}$$

The essential matrix has some interesting properties. Considering the matrices on expression 2.29, since S is a skew-symmetric matrix, *i.e.* $a_{ij} = -a_{ji}$ and R is an orthonormal matrix, their product is a well-known result in linear algebra, deduced by the Singular Value Decomposition.

Although the E matrix allows us to understand the relation of the points by its camera coordinates, it does not establish the relation of the pixel coordinates (q_l and q_r) that we are looking for. In order to achieve this relation, the intrinsic parameters of the two cameras must be introduced. Through equation 2.4, the relation $q = Mp \Leftrightarrow p = M^{-1}q$ is established and follows:

$$q_r^T (M_r^{-1})^T E M_l^{-1} q_l = 0. \tag{2.31}$$

The ***Fundamental matrix*** is thus introduced, setting up a relation between the pixel coordinates, *i.e.*:

$$q_r^T F q_l = 0, \tag{2.32}$$

where F is the *fundamental matrix* defined as $F = (M)_r^T E (M)_l^{-1}$.

2.4.2 Homography from two co-planar points

Homography establishes a relation between the physical coordinates of an object and its coordinates on the image plane. Homography can also be used to relate the image plane coordinates on a two-camera system. From Section 2.4.1 this relation was already obtained. However, it can not be used in our system, since it requires that the two used cameras are the same, *i.e.* have the same intrinsic parameters. Aiming to be able to perform a math transformation that allows relating the optical and thermal acquired images, the homography that relates two co-planar points - planar homography or perspective transformation - will be used. This relation is accepted when the two cameras capture the same plane from different angles. Therefore, the generated homography relation does not depend on the acquired image or the referenced plane under analysis.

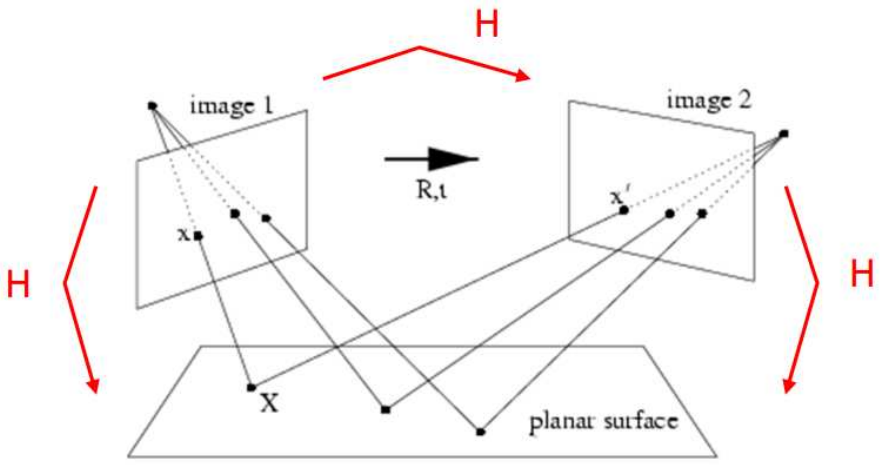


Figure 2.9: Schematic representation of projective transformation, taken from [10].

Let H be the homography matrix, expressed in equation 2.33.

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{2.33}$$

The relation between **x'** and **x** is translated as **x'** $= H\mathbf{x}$, where **x'** $= \begin{bmatrix} x' & y' & 1 \end{bmatrix}^T$ and **x** $= \begin{bmatrix} x & y & 1 \end{bmatrix}^T$. It is important to note that **x'**, **x**, and H are in homogeneous coordinates.

A relation between x' and x coordinates can be established as follows [11]:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{h33}} \text{ and} \tag{2.34}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{h33}}. \tag{2.35}$$

Since H is a 3×3 matrix, it has 9 degrees of freedom. By setting the condition $h_{33} = 1$ [11], it can become an 8 degrees of freedom matrix. Alternatively it can also be enforced by establishing the sum of all elements to 1 [11,12]. On this approach, the first condition will be used. By applying it and rearranging equations 2.34 and 2.35 one gets:

$$x' = h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' \text{ and} \tag{2.36}$$

$$y' = h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy'. \tag{2.37}$$

Combining for four distinct points seen from each camera leads to:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x'_4 & -y_4x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y'_4 & -y_4y'_4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{bmatrix} \tag{2.38}$$

In order to solve this equation, linear independence of all rows must be ensured. Therefore, at least three points of the groups $[(x_1,y_1), (x_2,y_2), (x_3,y_3), (x_4,y_4)]$ or $[(x'_1,y'_1), (x'_2,y'_2), (x'_3,y'_3), (x'_4,y'_4)]$ must not be collinear. This equation is then solved using least-squares estimation [13].

OpenCV provides the function *cv.findHomography()*, which returns the homography matrix H, from four distinct points and their corresponding points in each image under analysis. This algorithm also uses the condition $h_{33} = 1$. Therefore, when combining optical and thermal images, we can use this function to map the images and match the corresponding points. The OpenCV function *cv.warpPerspective()* applies the transformation matrix previously computed to an image and allows to visualize the result. Figure 2.9 already showed an example of a perspective transformation.

2.5 Temperature measurement model

Temperature measurements' reliability through thermal imaging techniques presents some challenges since the acquired temperature is influenced by the surroundings and depends on the distance between the camera and the object under observation.

All objects have temperature above 0K and emit thermal radiation which is described by Stefan-Boltzmann's law. Besides that, considering Kirchhoff's law [14], bodies also reflect a certain amount of energy, when exposed to incident radiation. Ideally, the temperature measurement should not be performed under any solar radiation, for instance. Furthermore, the surrounding environment also affects the measured value of the temperature, not only due to the atmospheric attenuation of the emitted radiation but also due to the surrounding objects and their emittance, which contributes to the global environmental radiation. Some types of systems for temperature measuring purposes need the direct intervention of an operator or a supervisor during the process of the data acquisition, due to the dynamics necessary during a measurement of this type (distance from the system to the object, angle of incidence, etc) [14].

Taking into account all the mentioned factors that influence temperature measurements, a model must be used in order to evaluate how the acquired temperature varies as the object gets further away from the camera, with the aim of getting the object's real temperature and to get more accurate results.

A model for the output signal of the camera's detector, considering three heat fluxes arriving on it is proposed in [15]. This model combines the flux emitted by the object under observation, φ_{ob}, the flux emitted by the ambient and reflected from the object, φ_{refl}, and the flux emitted by the atmosphere, φ_{atm}. The flux emitted or absorbed by the optical components and filters of the camera is considered negligible.

$$\varphi_{\text{ob}} = \varepsilon_{\text{ob}}(T_{\text{ob}})\tau_{\text{atm}}(T_{\text{atm}})M_{\text{ob}}(T_{\text{ob}}) \tag{2.39}$$

$$\varphi_{\text{refl}} = [1 - \varepsilon_{\text{ob}}(T_{\text{o}})]\varepsilon_{\text{o}}(T_{\text{o}})\tau_{\text{atm}}(T_{\text{atm}})M_{\text{o}}(T_{\text{o}}) \tag{2.40}$$

$$\varphi_{\text{atm}} = [1 - P_{\text{atm}}(T_{\text{atm}})]M_{\text{atm}}(T_{\text{atm}}) \tag{2.41}$$

The equations 2.39, 2.40, and 2.41 represent the fluxes mentioned in the previous paragraph. ε_{ob} and ε_{o} represent the emissivity of the object's surface and the band emissivity of the ambient, respectively. M_{atm}, M_{ob} and M_o are the radiant emittance of the atmosphere, object, and ambient, respectively. τ_{atm} is the band transmittance of the atmosphere. T_{atm}, T_{ob}, and T_{o} represent the temperature of the atmosphere, object under observation, and ambient, respectively, in K (Kelvin).

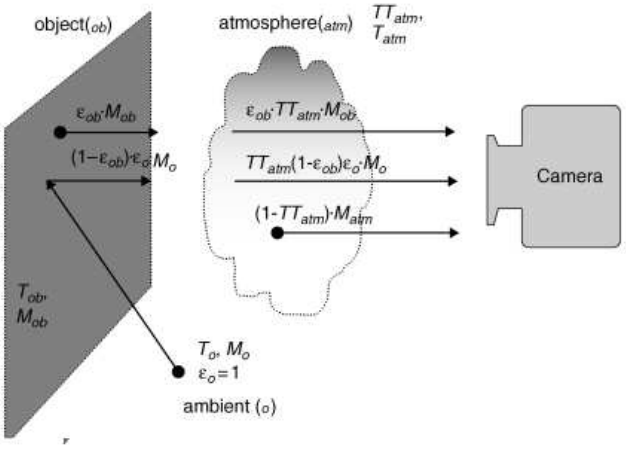


Figure 2.10: Heat fluxes and interactions model, taken from [15].

The detector's signal that express the total intensity of heat radiation arriving on the detector can be defined as [15]:

$$s = \varepsilon_{ob} \cdot \tau_{atm} s_{ob} + \tau_{atm} \cdot (1 - \varepsilon_{ob}) s_o + (1 - \tau_{atm}) s_o, \tag{2.42}$$

where s_{ob} is the signal corresponding to the total heat flux of the object under observation and s_{o} is a signal of a black body heat radiation at an ambient temperature, expressed as:

$$s_o = \frac{R}{e^{B/T_o} - F}, \tag{2.43}$$

where R, B, and F are constants obtained from the camera calibration fit, described in Section 2.5.1.

The signal of the object under observation, s_{ob}, can be obtained by combining the previous equations:

$$s_{\mathrm{ob}} = s \cdot \frac{1}{\varepsilon_{\mathrm{ob}} \tau_{\mathrm{atm}}} - \left[\frac{1-\varepsilon_{\mathrm{ob}}}{\varepsilon_{\mathrm{ob}}} \cdot \frac{R}{e^{B/T_{\mathrm{o}}} - F} + \frac{1-\tau_{\mathrm{atm}}}{\varepsilon_{\mathrm{ob}} \cdot \tau_{\mathrm{atm}}} \cdot \frac{R}{e^{B/T_{\mathrm{o}}} - F} \right] \tag{2.44}$$

The Atmospheric transmission, τ_{atm}, shows a dependency on the distance between the camera and the object. That can be modelled as [15]:

$$\tau_{\mathrm{atm}}(d) = exp\left[-\alpha \left(\sqrt{(d)} - \sqrt{d_{\mathrm{cal}}} \right) - \beta \left(d - d_{\mathrm{cal}} \right) \right], \tag{2.45}$$

where α and β coefficients depend on whether the camera is short-wave (SW) or long-wave (LW)[5]. For an SW camera α = 0.00393 $\mathrm{m}^{-1/2}$ and β = 0.00049 m^{-1} and for an LW camera α = 0.008 $\mathrm{m}^{-1/2}$ and β = 0 m^{-1} ([6]). These values were calculated based on the normal conditions of the atmosphere [15], considering T_{atm} = 15 ºC and relative humidity of $\omega_{\%}$ = 50%

Finally, in order to express an equation that translates the object's raw temperature, T_{ob}, in Kelvin (K), the equation 2.43 is rearranged to:

$$T_{\mathrm{ob}} = \frac{R}{ln\left(\frac{B}{s_{\mathrm{ob}}} + F \right)} \tag{2.46}$$

Therefore, T_{ob} can be expressed as a function of:

$$T_{\mathrm{ob}}(K) = f(\varepsilon_{\mathrm{ob}}, T_{\mathrm{atm}}, T_{\mathrm{o}}, \omega, d) \tag{2.47}$$

It must be reinforced that the model above is a simple representation of an infrared object's temperature measurement. A more complex model is available in [15], considering other sources of heat radiation flux.

2.5.1 Temperature calibration fit

The system's calibration is done considering a short distance between a technical black body and the camera. In such conditions, the values $\varepsilon_{\mathrm{ob}}$ and τ_{atm} can be assumed ≈ 1. Therefore, from Equation 2.44, follows $s_{\mathrm{ob}} = s$.

[5]This concept refers to the wavelength spectrum, which is based on the length of the wave. For instance a SW is tipically defined as 0.9 - 1.7 μm wavelength range, and LW is typically defined as the 8.0 - 14.0 μm wavelength range. [16].

[6]The camera used in the developed system is a LW camera

Each signals s_i measured by the camera corresponds to different temperatures T_i, set on technical black bodies [7], where i represent each done acquisition:

$$s_i = \frac{R}{exp(B/T_i) - F}, \tag{2.48}$$

where R, B and F are constants obtained for the best fit function. The values of R, B and F depend on the camera model and on the temperature operation range [15].

2.5.2 Temperature compensation model

The measurement of temperatures at long-distances can be compensated by evaluating the influence of the contrast and the atmospheric transmit ranges on the temperature measurement accuracy [17, 18].

When measuring an object's temperature through IR cameras, there is a significant amount of background radiation reaching the camera's sensor that affects the measurement accuracy.

This model first considers that the object under temperature measurement is a black body. The total temperature measured by a IR camera, T_{ob}, is a sum of two parts: the temperature of the tested object and the temperature of the environmental backgrounds:

$$T_{ob} = yT_B + zT_a, \tag{2.49}$$

where T_B is the real blackbody temperature and T_a is the atmospheric temperature. The values y and z are weights of the proportion of the object temperature and the environmental background temperature, respectively, and have the relationship $y = 1 - z$. The factor y will change over the distance, therefore there is a explicit dependency on the distance.

Taking the previous relation between y and z, the equation 2.49 can be rewritten as:

$$T_o = yT_B + (1 - y)T_a. \tag{2.50}$$

As the difference between T_{ob} and T_a gets smaller, the smaller is the factor y. As a consequence, the measurement uncertainty increases.

To evaluate the evolution of y as the distance increases[8], the acquired data is fitted to a model considering $y = f(d)$ [17]. Once having the factor y as a function of the distance and having

[7] For this model [15], a set of technical black bodies were used. A technical black body is a device that simulates the properties of a black body, and has a certain temperature associated with it.

[8] The y factor is defined on [17] as the ratio of the value measured by non-compensated infrared temperature measurement system (the acquisition on the RoI from MLX90640) and to the actual temperature, in this study usually measured with an TMPx75 sensor [19].

the estimated distance, the estimated temperature of the object under temperature measurement is given by:

$$\hat{T} = \frac{T_{ob}}{y}. \tag{2.51}$$

2.6 Object detection algorithms

To meet the system requirements, the system must be able to detect people or objects. Thus, object detection algorithms should be introduced. The detection algorithm will be incorporated in the system in order to detect regions of interest and measure their temperature, through the subsequent correspondence of optical and thermal images. This section will focus on the Haar-cascade classifiers. Although there are more advanced methods based on deep learning, such as CNN (Convolutional Neural Networks), the use of the chosen algorithm is sufficient, in particular because it was already implemented in the used software OpenCV.

2.6.1 Haar-cascade classifiers

Paul Viola and Michael Jones developed a robust and extremely rapid machine learning algorithm for object detection [20]. In this approach, Haar features were used. In this work they proposed a new image representation, the **integral image**, that improved the time of feature evaluation. Following this work, a classifier was also developed by using AdaBoost [21] in order to also improve the classification time. A more complex classifier combination in a cascade structure was also proposed [20].

Haar features

When computing algorithms for object detection, there are many advantages to using the image's features rather than the direct values of the individual pixels, mainly due to their operating speed and their information limitations.

An Haar-like feature can be described as a rectangle that contains a light side and a dark side [22]. There are three types of Haar-like features (figure 2.11 [23]: *two-rectangle feature*, *three-rectangle feature* and a *four-rectangle feature*. As indicated in the paper, a two-rectangle feature is the difference between the sum of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent. A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle.

Finally a four-rectangle feature computes the difference between diagonal pairs of rectangles [20].

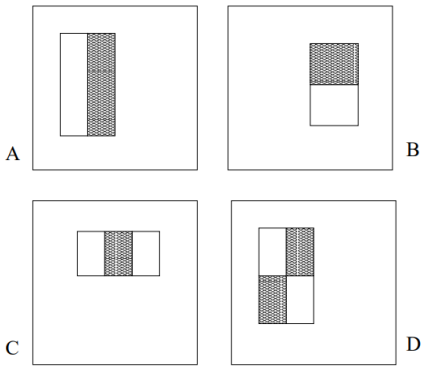


Figure 2.11: Types of Haar-like features. Two-rectangle features are shown in (A) and (B). (C) shows a three-rectangle feature, and (D) a four-rectangle feature, taken from [20].

Integral image

The integral image is an intermediate representation of the original image, where the rectangle features are computed. By definition, at the location (x,y) the integral image contains the sum of the pixels above and to the left. Equation 2.52 describes this relation, where $ii(x,y)$ is the integral image and $i(x,y)$ is the original image.

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \tag{2.52}$$

Computing the integral image over the original image can be done in one iteration, by using the following recurrence expressions:

$$s(x,y) = s(x,y-1) + i(x,y) \tag{2.53}$$

$$ii(x,y) = ii(x-1,y) + s(x,y) \tag{2.54}$$

Haar-like features only provide three types of orientations: vertical, horizontal, and diagonal. Although these features may seem limited due to their limited flexibility, by combining them with the integral image approach, the learning process becomes very effective.

Learning mechanism

The classifiers are trained with a big set of positive images and negative images[9] of the object to be detected. Once this set is prepared, the features need to be extracted from it. Applying the Haar algorithm described above, using a variant of *AdaBoost* algorithm. The final classifier is a weighted sum of weak classifiers [20]. Each weak learner determines the optimal threshold classification function, based on the selection of the Haar-like feature and its positive or negative results. A single weak classifier is not enough to classify the image, therefore it is combined with many others in order to build a strong classifier. On the published article, Paul Viola and Michael Jones show that from a frontal face classifier, built with 200 features, a detection rate of 95% was obtained with a false positive rate of 1 in 14084. For a 384 by 288 pixels image, it took 0.7 seconds to perform. Although the results have improved, the classifier's performance does not satisfy in many real applications because of its time latency.

Cascade classifiers

A cascade of classifiers was developed in order to improve detection performance and reduce computation time [20]. Considering that most of an image is a non-face region, the cascade classifiers rely on a simple method based on a degenerate decision tree operating at different scale sizes of the image, that is called as *cascade*: "A positive result from the first classifier triggers the evaluation of a second classifier which has also been adjusted to achieve very high detection rates. A positive result from the second classifier triggers a third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the sub-window. [20]". AdaBoost-trained classifiers are then used to build the stages of the cascade, which are then adjusted to reduce the false negatives.

Training Haar-cascade classifiers with OpenCV

OpenCV already provides some trained classifiers for object detection and facial recognition. However, any user of a system can train a classifier to be able to recognize a particular object under study.

As mentioned above, to train a classifier it is necessary to have a set of positive and negative samples of an object, that is, to have images that contain the object and others that do not. These images are labeled images with bounding boxes, whose coordinates are stored in a file.

[9]The positive images are the images that contain the object under recognition. Those who do not contain the object are called negative images.

After having the samples set completed, the cascade is trained considering a set of common arguments: a file where the data will be stored, a vec-file[10] containing the positive samples, a file containing the background (negative samples), the number positive and negative samples used in training for every classifier stage, and the number of cascade stages to be trained. For the cascade parameters the types of the stages and of the features, the width and height of the training samples must be defined. For the boosted classifier parameters, the type of the boosted classifier is defined, and the minimal desired hit rate and the maximal desired false alarm rate for each stage are defined. It must be specified whether trimming is used and it weight, usually set to 0.95. The maximal depth and the maximal count of the weak trees are also set. Finally, the type of the Haar-like feature parameters is defined [24].

[10]A vec-file is a binary image file which contains images.

3 Experimental setup

3.1 System design and parts

To acquire images and measure an object temperature, two types of cameras will be used: thermal and optical cameras. A control and processing unit is needed in the system, because there must be a way to process the information that is acquired. The equipment chosen to perform the processing was a RaspberryPi (version 4B+). The cameras used were an Adafruit MLX90640 Thermal camera and a RaspiberryPi Camera Module V2. Figure 3.1 shows how the cameras are connected to the Pi.

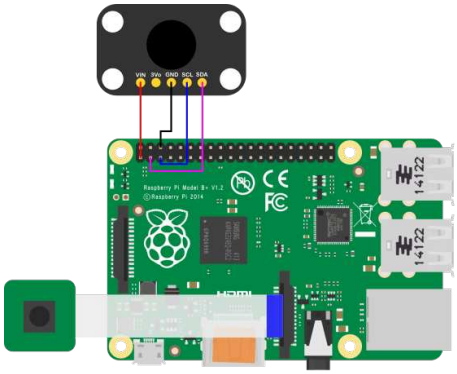
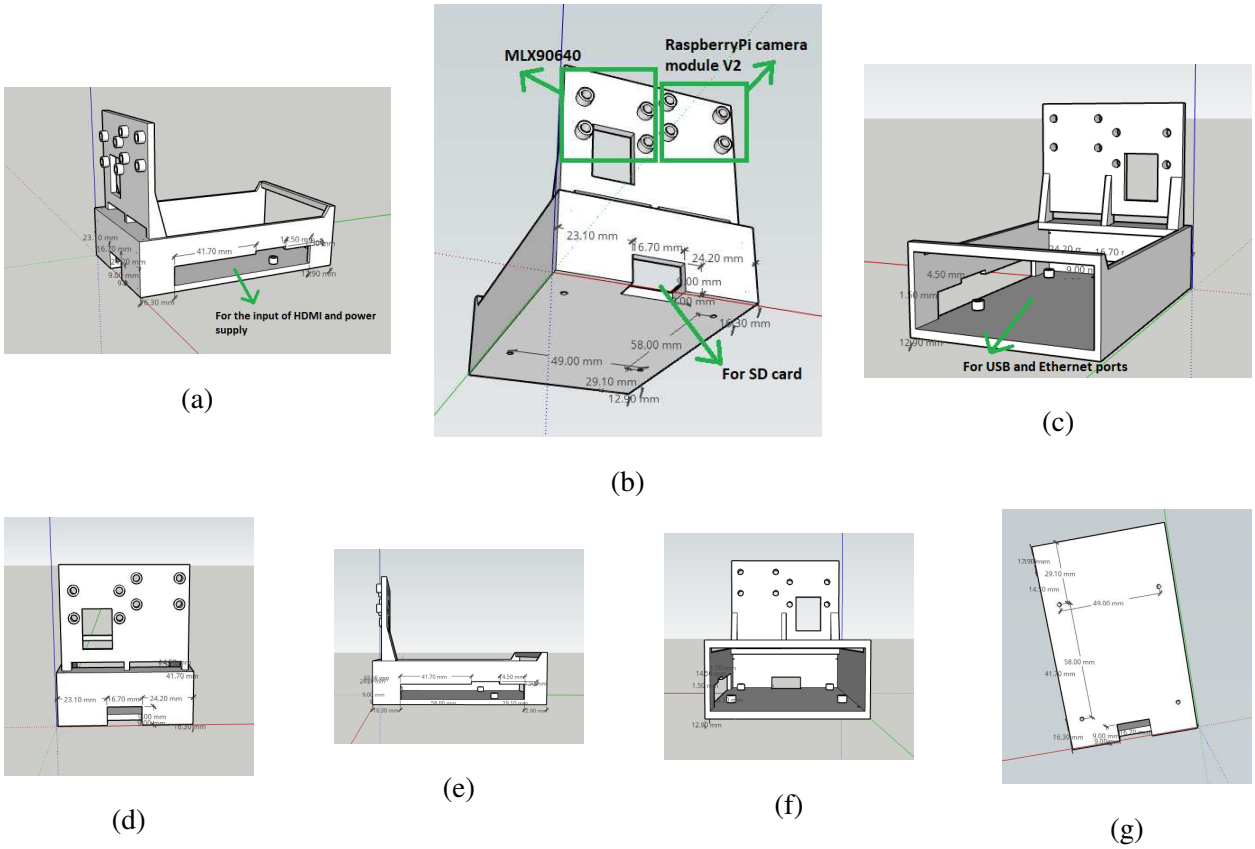


Figure 3.1: Experimental setup showing the RaspberryPi and the two cameras. The RaspberryPi illustration is taken from [25]. This image is merely illustrative, since the represented RaspberryPi does not correspond to the one used in the system.

The cameras are in a fixed position sustained by the system's case, designed for this purpose. The case was designed using the Sketchup software [26]. Figure 3.2 shows an overview of the sketched case designed for the system.



(a) (b) (c) (d) (e) (f) (g)

Figure 3.2: Different views of the system's case

3.1.1 Cameras

RaspberryPi camera

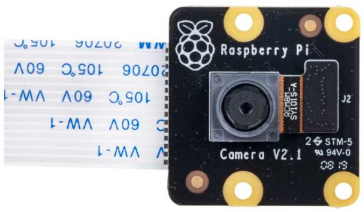


Figure 3.3: RaspberyPi camera, image taken from [27]

The RaspberryPi camera (Camera Module v2 [28]) is an 8Mp camera and has a 3280x2464 sensor resolution. The camera connects with the RaspberryPi via the CSI connector, specially designed for this type of camera. The maximum frame rate is 30fps for 1080p definition and 60fps for 720p. This camera is not equipped with an IR filter, which means that it is also sensitive to radiation near IR spectrum.

Although RaspberryPi camera has a very high resolution, we will acquire images with 640x480 resolution, in order to facilitate the correspondence between the optical and thermal images. This

is done by simply choosing the image width and height, 640 and 480, respectively.

Adafruit MLX90640

Figure 3.4: Adafruit MLX90640, taken from [29]

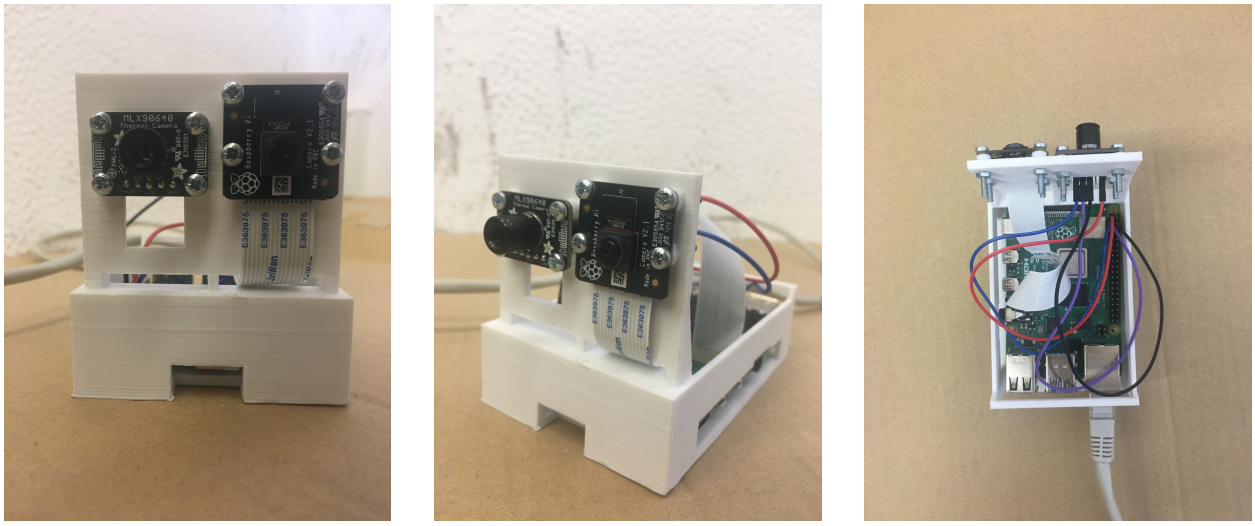
The Adafruit MLX90640 is a 24x32 array of IR thermal sensors, corresponding to 768 individual pixels. The camera is able to measure temperatures from -40ºC to 300ºC, and has an accuracy of $\pm$ 2ºC, in the range of 0ºC-100ºC. Its FOV (Field of View) is 55º$\times$35º. It communicates with the RaspberryPi through the I2C protocol and SDA/SCL pins. Its maximum frame rate is 16Hz [29].

The pixel sensors acquire the thermal energy emitted by objects and that is then converted to a signal by the camera electronics. Once the conversion is completed, the signals from all pixels are readout and processed at the RaspberryPi, which is able to produce an image, by applying a colormap to the acquired data.

Since the RaspberryPi camera has a much higher resolution when compared to MLX90640, some adjustments are needed in order to proceed with the image analysis. When referring to scaling up the images, it is possible to resize the obtained thermal image from 24x32 to a 640x480 resolution, equal to the resolution obtained for the RaspberryPi camera. Through a process of interpolation, a new bigger image is produced. Therefore, once the cameras are calibrated, optical and thermal acquired images match at every pixel.

3.1.2 Assembly of the experimental setup

The images on Figure 3.5 show the final assembly of the experimental setup of the system.



(a) (b) (c)

Figure 3.5: Experimental setup of the system, composed by a Raspberry Pi and the two cameras, on the holding case

3.2 Heatsink

For many IR analysis applications presented in this work, an anodized aluminium heatsink [1] was used (see Figure 3.6 and 3.7), with an emissivity ε=0.85 [30], heated by a resistor supplied with 30 V voltage. At this supplied voltage the heatsink usually reaches its thermal equilibrium, at $\sim 50C$. Once the heatsink was on thermal equilibrium, the acquisitions were made. The temperature of the heatsink was monitored with a TMPx75 sensor [19].

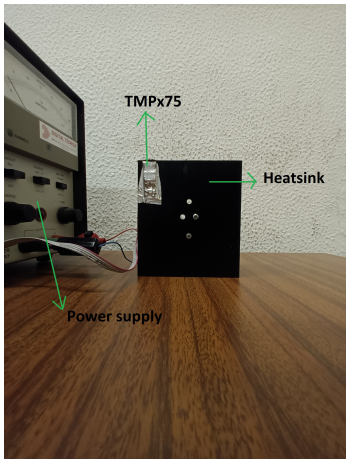


Figure 3.6: Anodized aluminium heatsink. On the top left corner of the heatsink there is the TMPx75 sensor, used to measure the temperature of the heatsink.

[1] The used heatsink is a very good thermal conductor, which allow to have a homogeneous temperature through all its surface

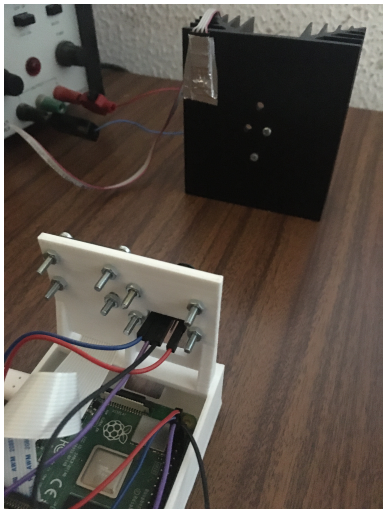


Figure 3.7: Setup for the heatsink's temperature analysis.

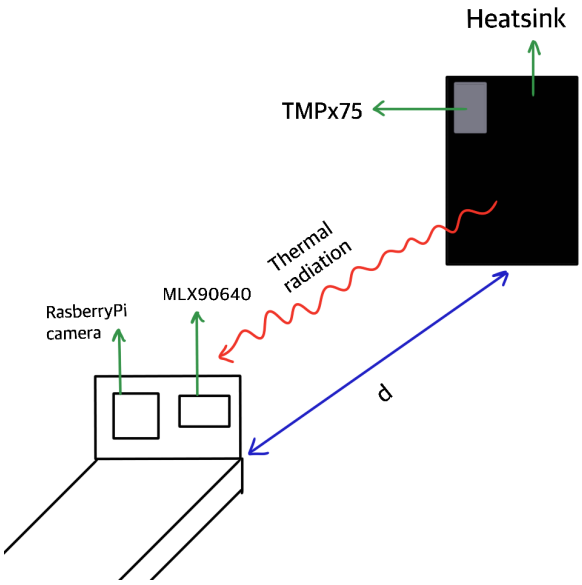


Figure 3.8: Schematic representation of the setup for the temperature measurement of the heatsink

3.3 Connecting to RaspberryPi

In this system it was not used a direct way of displaying the RaspberryPi Desktop. Instead, the RaspberryPi was acessed via SSH connection.

3.4 Acquiring data from both cameras

Both cameras have distinct ways of acquiring data. It is necessary to combine this techniques in order to perform the algorithm. The combination was necessary to fully aggregate their potentialities.

The MLX90640 driver [31] contains some procedures and functions necessary to use the camera in the system. For the performance of this camera, these procedures were take into account to initialize the camera and process its data. Combining with other OpenCV tools, such as color mapping, to help image visualization, the IR images were visualized in a real-time application, and acquired when required by the user.

The RaspberryPi camera was simpler to use, since it is already a module of RaspberryPi. The characteristics that were pre-defined were brightness, contrast, saturation and gain of the image obtained.

4 Results and discussion

4.1 Understanding the MLX90640 behaviour

4.1.1 Defining the color map

The output matrix from the MLX90640 implemented algorithm is a grayscale image, whose values are on the range 0-255. To help visualisation, a color map is applied to the intensity matrix, from the relative intensities on the image matrix. It was important to define the minimum and the maximum temperatures for our test conditions, in order to being able to compare different images, under the same color range constrains and scale. Therefore, the minimum temperature was fixed at 20ºC, that corresponds to a colder color (dark blue - minimum intensity (0 on a gray scale)) and the maximum temperature was fixed at 50ºC, which corresponds to the hotter color (dark red - maximum intensity (255 on a gray scale)). As a consequence, an intensity of 8.5 represents a degree, on the 0-255 scale.

4.1.2 Stabilization time

Evaluating stabilization time of the IR camera is very important for understanding our measurements. For example, if the stabilization time takes too long, it would be unwise to acquire data right after the electronics turn on.

To study the stabilization time of the IR camera, the temperature of a heatsink (see Section 3.2), previously brought to thermal equilibrium was acquired from a distance of 50cm during 5 minutes right after the electronics was turn on. The results are presented in figure 4.1. From this figure, it can be seen that there is no significant sistematic changes on the acquired temperature, which suggests that the stabilization time is much shorter than the acquisition period, so it can be neglected.

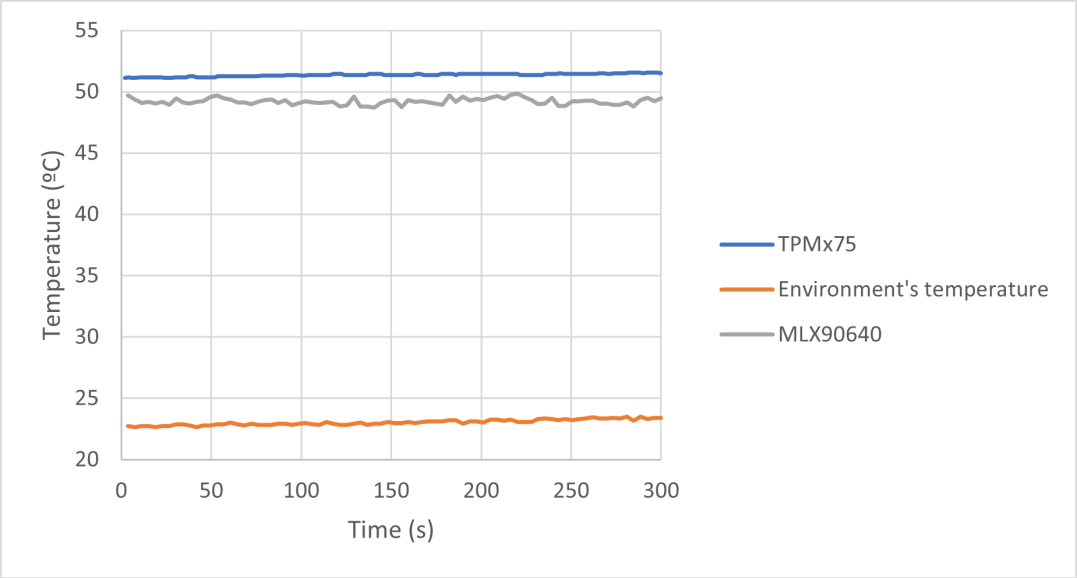


Figure 4.1: Acquisition for evaluating stabilization time of the IR camera. The blue line refers to the measured temperature of the heatsink, acquired with an TMPx75 sensor [19]; the gray line shows the acquired temperature from the MLX90640 matrix - the value represents an average value of the selection of the region of interest; the orange line corresponds to the environment's temperature also measured by the MLX90640 camera.

4.1.3 Camera calibration

Since the MLX90640 datasheet does not provides any information about the camera calibration[1], a study was made in order to understand the conditions assumed for that calibration.

Following from Section 2.5.1 it is necessary to calculate the object's real temperature based on the camera's output signal. It is not possible to get the raw signal[2] s (see Equation 2.42) from the camera's sensor. The output signal of the MLX90640 camera is a 24x32 temperature matrix. Therefore this matrix was assumed to be a raw output signal, and it was compared to a control temperature measured by other sensor. This assumption still satisfies the condition 2.48, since it is relating a real object's temperature as an input and a temperature (signal) measured by the camera

[1]It is referred by [15] that a calibration is done at each power-on. From that automatic calibration the best-fit R, B, and F parameters are internally calculated. For these calculations, some parameters must be assumed, and are determined in specific conditions, defined by the manufacturer. However, the conditions under which they are carried out are not very explicit. Furthermore, and since the camera datasheet does not provide any information about this calibration, and that the calibration parameters are different for each type of camera, we should look for the manufacturer's calibration conditions.

[2]The raw signal follows an approximation of the integral of the product of the black body radiant exitance $M(l,T)$ incident on the detector, according to Planck's law, and function $S_k(l)$ describing the relative spectral sensitivity of the camera [15].

as an output.

The body under test was a heatsink, considered a black body by approximation[3]. For this measurement, the distance between the camera and the heatsink was set to 2cm to minimize the influence from the surroundings and guarantee that the heatsink completely fills the camera's FOV in order for all pixels to see the same temperature. Then the average value of all 24x32 pixel outputs were calculated for different values of the temperature of the heatsink and the results fitted using 2.48 The heatsink temperature range considered for the test is a typical range for human's temperature. A TMPx75 [19] sensor was used to monitor the real temperature of the heatsink, having an accuracy of ($\pm 1C$). The results from the fit are shown in Figure (4.2).

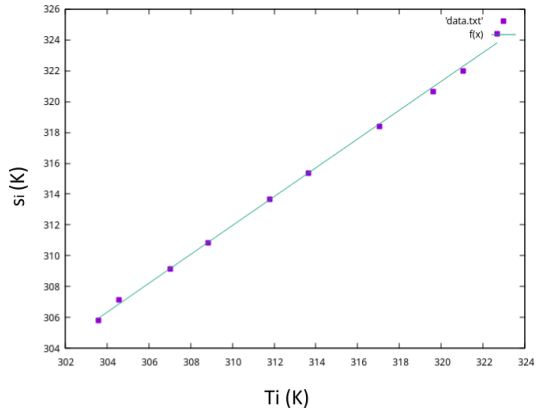


Figure 4.2: Temperature of the object vs the signal from camera. The fitted values for the parameteres in Equation 2.48 are R=1.45 ($\pm$ 66.61), B=1.33846 ($\pm$ 62.18), F=0.99967 ($\pm$0.009624).

From the figure 4.2 we see that the model fits well our data. However, it presents degeneracy: all the values of the correlation matrix from the fit are 1. Therefore, the presented fit does not provide relevant information for the calibration process, contrary to what was expected. This may suggests that the internal parameters R, B and F are already determined, under a calibration done by the manufacturer. To confirm this hypothesis, we will proceed with the analysis of calculating T_{ob} based on the acquired value s

Considering the the R, B, and F values from the fit on expression 2.48, and proceeding with the analysis, the heatsink's temperature was measured through various distances. Considering the equations 2.44-2.46, for s_{ob}, τ, and T_{ob}, respectively, the value of T_{ob} was calculated. The results are presented in Figure 4.3. Although the camera datasheet does not provides information about the wavelength operation range (whether its detector is a SW ($\sim$+690ºC - $\sim$+1800ºC) or LW ($\sim$-80ºC - $\sim$+90ºC) T_{ob}, from temperature operation range of the camera (see Section 3.1.1) it is possible to deduce that the camera is an LW camera. Therefore, the LW analysis was done (Figure 4.3).

[3] Although this approximation is made, it can introduce errors, since the real emissivity of the heatsink is $\varepsilon = 0.85$

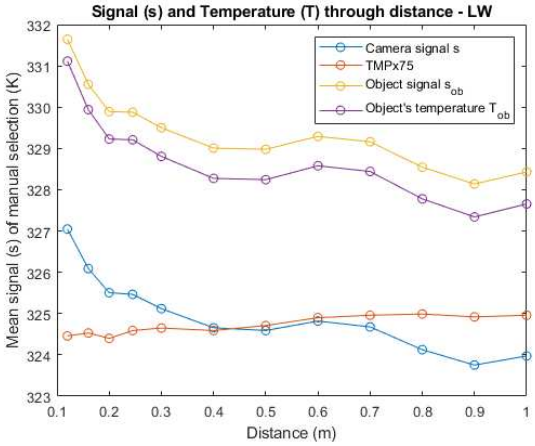


Figure 4.3: Calculating T_{ob} considering a LW-detector

From Figure 4.3, it can be seen that the obtained T_{ob} is much higher than the acquired signal. According to the previous analysis, in which the degeneracy was verified for the values, and taking into account the analysis between s and T_{ob}, leads to conclude that the camera is already calibrated, and the camera's given signal is an already corrected value for the temperature object, T_{ob}. To support this affirmation, verifying the MLX90640 driver, we see that the algorithm for temperature acquisition already considers the reflected temperature on the environment, and that the camera has an internal sensor that measures its temperature.

Besides that, it can be seen that at short distances the temperature measured by the camera selection is higher than the control temperature measured by the TMPx75 sensor. This suggests that it is possible that there is a defined a calibration distance by the manufacturer. This also will be part of our discussion later on section 4.1.6.

4.1.4 Noise RMS vs Refresh rate

The camera datasheet [29] suggests that there is a influence of the *refresh rate* on the noise performance. Before introducing the thermal analysis, in this section an evaluation of the influence of the *refresh rate* on the noise RMS was made. To evaluate this relation, an heatsink (see Section 3.2), with emissivity $\varepsilon = 0.85$ [30] previously brought to thermal equilibrium was under observation. Through its IR image, a region of interest was manually selected. The temperature of the object was then given by the average temperature of all pixels on the made selection. The noise RMS was considered as the standard deviation on the selected region of the thermal image.

Seven acquisitions were made, on laboratory conditions, using an heatsink (see Section 3.2) in thermal equlibrium, at a room temperature of $T_{room} = 25°C$, defining at each acquisition the frames per second (FPS) at 1, 2, 4, 8, 16, 32 and 64, respectively. From the number of FPS, the algorithm

sets a refresh rate. For temperature control, a TPMx75 [19] was used. All acquisitions were made at a object-camera distance $d = 50$cm.

The table 4.1 shows the results of this analysis. The noise RMS increases as the FPS increase too. Although the noise RMS increases, there is not a significant change on the average measured temperature of the selection, acquired by the camera. Therefore, $FPS = 32$ was considered as a standard value for our measurements[4]. If it was ever not considered, it will be mentioned.

FPS	TMPx75 (ºC)	MLX90640 (ºC)	Noise RMS (ºC)
1	49,25	50,91	0,56
2	49,31	50,91	0,61
4	49,16	50,97	0,65
8	49,06	51,05	0,66
16	49,25	50,82	0,74
32	49,31	50,68	1,10
64	49,38	51,25	1,17

Table 4.1: Results of the influence of FPS on Noise RMS

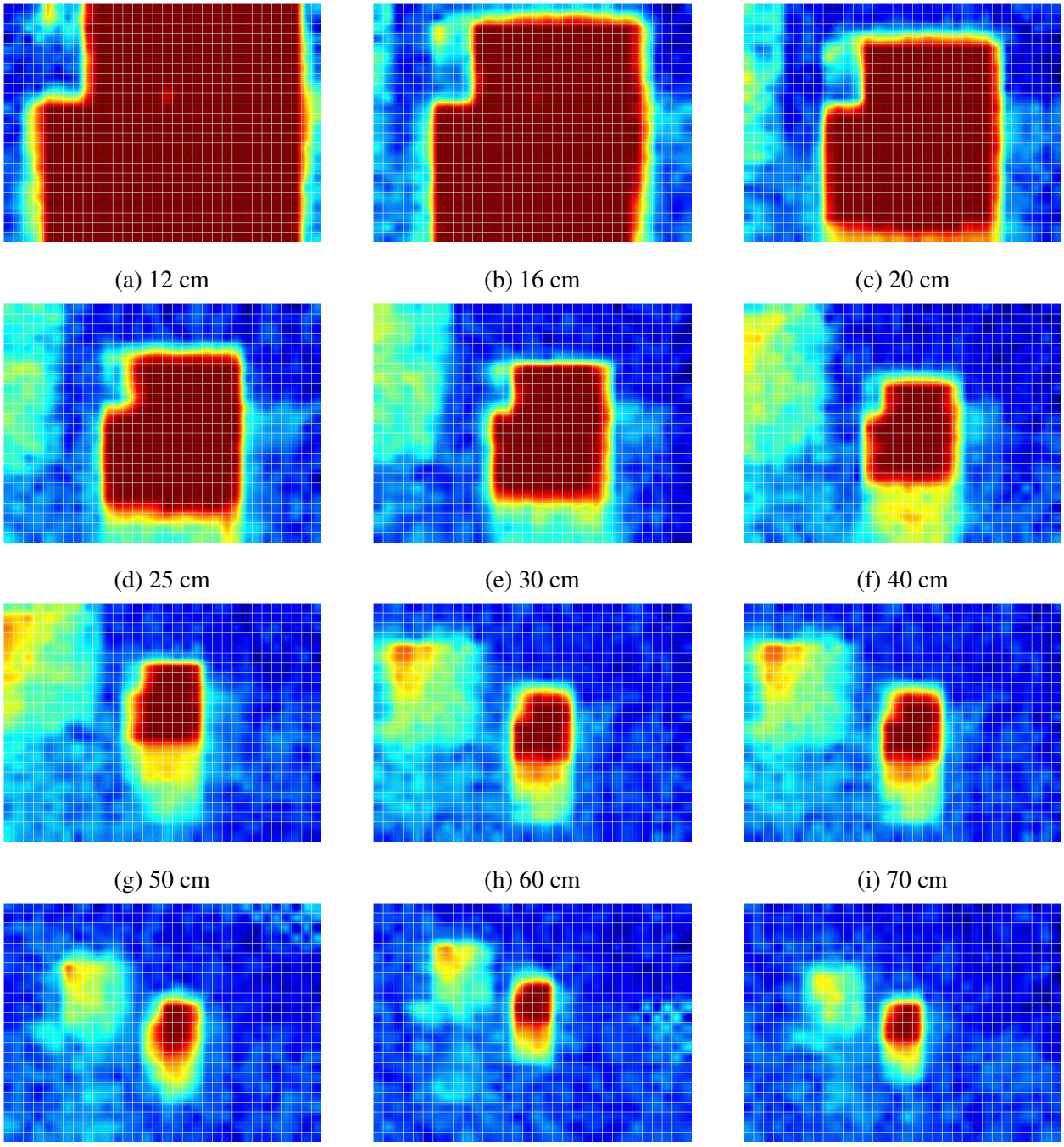
4.1.5 Temperature acquisition analysis

As mentioned in Section 2.5.2, when measuring an object's temperature through IR cameras the influence of environment radiation must be considered. The further away the object is from the camera, the greater the influence of the surroundings on the temperature measurement. In addition, the signal s_{ob} (eq. 2.44) gets weaker due to the losses to the atmosphere and thus the estimated T_{ob} (eq. 2.46) decreases. In order to evaluate the MLX90640 camera's performance through various distances, a system with an anodized aluminum heatsink (see Section 3.2) was used. The temperature acquired with the TMPx75 sensor was then compared to the acquired camera temperature measurement. This experiment occurred at a room temperature of $T_{room} = 26°C$. The setup of this experiment can be seen on figure 3.7

The acquired images can be seen in Figure 4.4. The selection of the RoI was done manually. The selection was done by trying to avoid the object edges, only selecting an one-color full-colored pixel, *i.e.* the object must fulfill the entire FOV of the selected pixel, in order to measure the

[4]It was chosen to maintain the acquisition conditions with the previously taken data. Moreover, if using for instance 1FPS the system would be too slow for a person's moving face.

temperature with the maximum sensitivity [29]. Furthermore, the larger the distance is, it becomes more difficult to guarantee the previous requirement increasing the measurement error.



(a) 12 cm (b) 16 cm (c) 20 cm

(d) 25 cm (e) 30 cm (f) 40 cm

(g) 50 cm (h) 60 cm (i) 70 cm

(j) 80 cm (k) 90 cm (l) 100 cm

Figure 4.4: Thermal images of the heatsink taken from various distances. As the distance increases, the object's appearence becomes smaller, being more difficult to define its boundaries, and consequently the edges effect increases.

As it can be seen in Figure 4.4, there is a colder area in the upper left corner of the heatsink. This is the region where the TMPx75 sensor was placed. For measurement purposes, the part of that area was not considered and the selection was done as homogeneous as possible, for the reasons previously explained.

The temperature of the object is then given by the average temperature of all pixels on the selection. The noise RMS is considered as the standard deviation on the selected region of the thermal image (see on equation 4.1), considering the manual selection a matrix $M_{n\times m}$, with the top left corner with coordinates (i, j) and bottom right corner (k, l) on the image coordinates, with N elements (where $N = n \cdot m$). The μ value is the arithmetic mean of all $M_{n\times m}$ pixel values.

$$\text{Noise RMS} = \sqrt{\sum_{i,j=1}^{k,l} \frac{(M_{(i,j)} - \mu)^2}{N}} \tag{4.1}$$

The results of the mean value and standard deviation measured for a manually defined RoI are presented in the figure 4.5.

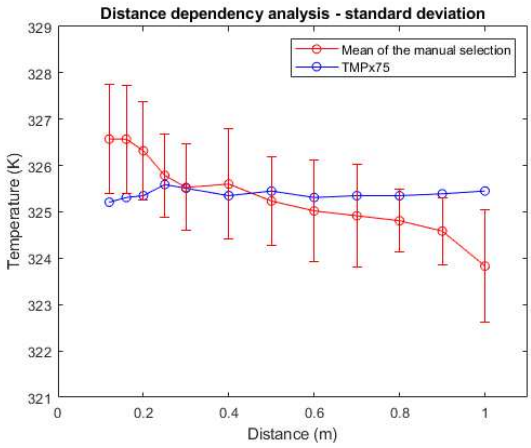


Figure 4.5: Comparison on the temperature measured by the camera (using a manual RoI selection) and by the TMPx75 sensor. The temperature of the RoI decreases as the distance improve, due to the effect of the evironment's temperature. The error bars represent the noise RMS on the acquisition

In figure 4.5, as the distance increases the object's appearance gets smaller and its selection window size decreases. The noise RMS on the selection stays relatively constant through all the measured distances, being approximately $1K$. This value agrees with the noise RMS quoted by the manufacturer to a refresh rate of 32Hz [29] (see on Section 4.1.4).

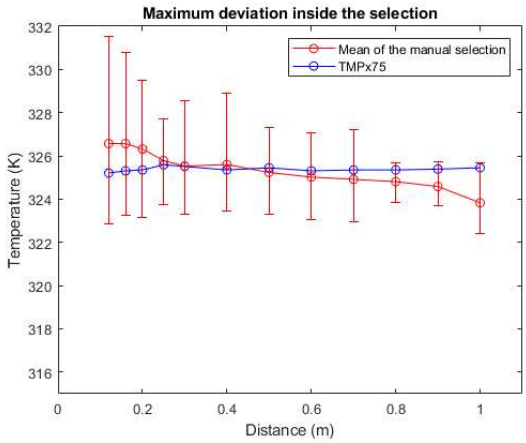
Uncertainties

While the uncertainty represented in figure 4.5 refers to the noise RMS of the RoI, the error bars represented in figure 4.6 shows the deviation on the selection and in the neighbourhood of the selection. In Figure 4.6a, the maximum deviation is calculated considering equations 4.2 and 4.3, where $M_{n\times m}$ is the matrix of the RoI and μ is the arithmetic mean of all $M_{n\times m}$ pixel values.

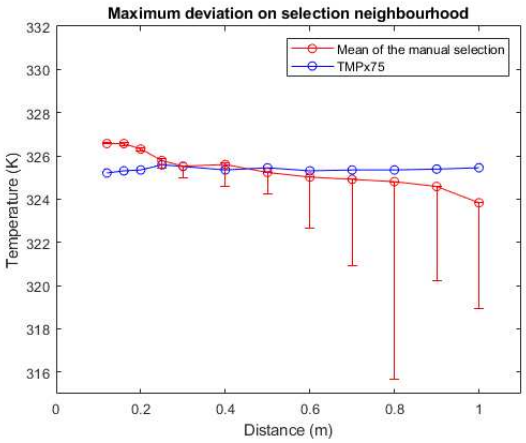
$$\text{Maximum positive deviation} = |\mu - max(M_{n\times m}))| \tag{4.2}$$

$$\text{Maximum negative deviation} = |\mu - min(M_{n\times m}))| \tag{4.3}$$

Similarly, on figure 4.6b the maximum deviation is calculated based on a sliding window moving by one pixel in every direction around the selection, looking for eight other selection combination in the selection's neighbourhood. Since the selection is done manually by an user, this method provides a proxy to account for the errors the manual selection may introduce in the analysis. That is why it was chosen to do the analysis on figure 4.6b.



(a) Maximum deviation inside the selection



(b) Maximum deviation on the sliding window selection

Figure 4.6: Deviation analysis

In Figure 4.6 (a) it can be seen that the smaller the distance is, the higher the maximum deviation is. At closer distances the pixel area selected is much bigger and therefore the expected deviations are bigger too[5]. Though the noise RMS is relatively constant through all the considered distances, when at larger distances, the overall pixel selection present a more homogeneous behaviour, and the amplitude of the deviation decreases. From these bars we can also infer that the results may not have a high precision, due to statistical fluctuations. The error bars have different size in both positive and negative directions, however for every distance measured the values seem very close.

In contrast, in figure 4.6 (b), at closer distances the error on the selection is negligible. Since at closer distances the selected area is much bigger, the neighbourhood still conserves the same temperature range of values. Then, the result of the small error bar at closer distances is expected. As the distance increases and the area decreases, the object's appearance becomes smaller (see in Figure 4.4) and it is more difficult to define the object's limits. This happens because the sensibility decreases since the object does not fulfill the FOV of each individual pixel, and because the increasing importance of the surroundings in the measurement, which are at different temperature from the heatsink, as expected. Besides that, we see that the deviation occurs mainly on the negative direction. The contrast between the heatsink and the environment (increasing of the edge error) becomes more relevant as the distance increases, and therefore the measured temperature decreases. Furthermore, since there is no positive error bar from distance 0.60m, this leads to conclude that the selected RoI is the highest effective temperature for the temperature measurement, *i.e.* any other selection on its neighborhood would lead to much lower temperature measurements.

4.1.6 Distance dependency analysis

It is clear from figure 4.5 that there is a strong dependency with the distance, with the temperature measured by the camera decreasing as the object-camera distance increases, affecting the system's accuracy.

At close distances the effects of the environment's on the camera measurement should be neglected, since the effect of the atmosphere is much shorter (see Figure 2.10). Therefore, it was expected that at closer distances, the temperature acquired by the camera would be as close as possible to the real temperature, acquired by the TMPx75 sensor. However, for shorter distances the measured temperatures by the camera shows up a little higher when compared to the TPMx75

[5]When moving the heatsink away from the camera, the RoI area decreases, *i.e.* the number of pixels that contribute to the arithmetic mean measurement also decreases. The deviation from the mean can be smaller because the mean will be more deviated towards the outlier.

ones. In order to understand this behaviour, this procedure was repeated several times with consistent results. Thus, for short distances we must consider a bias on the measurement. This also may suggests an optimal distance operation range, as mentioned previously on section 2.5.1. Therefore, an operation range will be defined, based on our results, in consistency with the internal calibration of the camera.

Defining a distance operation range

From the previous analysis, there was a bias on the measurements for distances shorter than 50 cm. Due to this, the optimal operating range of the system was established from 50 cm to 100 cm. The 100 cm limit is mainly imposed by the fact that the system is limited by its connection wires, and also because at greater distances the size of an object in the corresponding thermal image becomes very small and difficult to analyze. All analyzes that will be presented next will be studied in this operation range.

4.1.7 Compensation model

Since the optimal distance operation range is already defined, there is still the challenge regarding the compensation of the environmental effects on the temperature measurement. In Section 2.5.2 a model for temperature compensation was proposed, considering that the acquired temperature is a combination of two weighted temperatures: the real object's temperature and the atmosphere temperature. In this section, this model was tested and an evaluation was done in order to understand if it improved the measurements in order to increase their accuracy.

Focusing only on the optimal distance operation range, defined as $50\text{cm} - 100\text{cm}$, the y factor was calculated for several distances and the data was fitted with a line. Figure 4.7 shows the measured points together with the fitted line $y = -0.0076d + 1.0037$ with an $R^2 = 0.8015$. Besides, the point at $1m$ could be considered an outlier, taking into account a linear fit. Instead, other fit model could be used instead of linear fit, in order to increase the R^2 value.

Applying the y factor on the measured temperature, the results presented on Figure 4.8 were obtained. The systematic error related to the distance effect no longer exists.

Table 4.2 shows the MAE (Mean Absolute Error) (equation 4.4) before and after applying the compensation model. From this table it is shown that with the compensation model the MAE decreases significantly, which means that the compensation model increases the measurement accuracy.

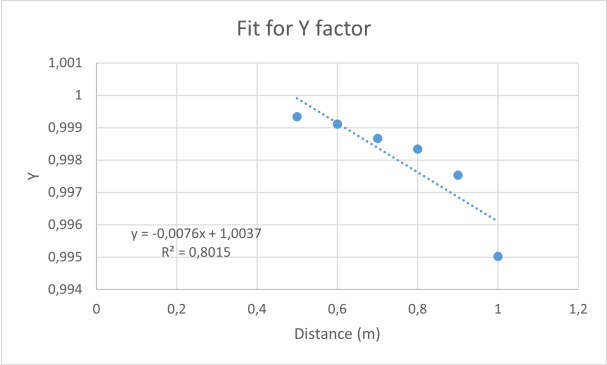


Figure 4.7: Fit of the factor y vs distance - see Section 2.5.2

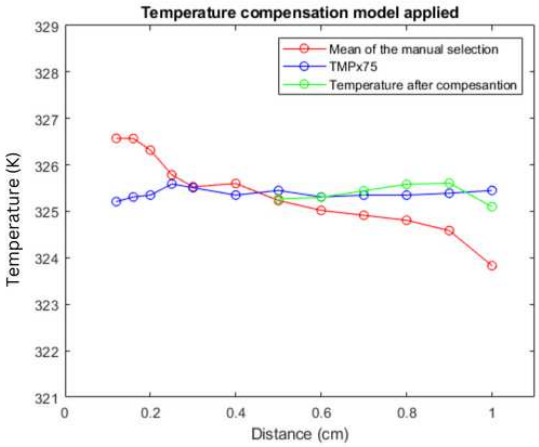


Figure 4.8: Application of the temperature compensation model (green line)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.4}$$

	MAE
Before compensation	0.7
After compensation	0.2

Table 4.2: Mean absolute error before and after temperature compensation

4.1.8 Critical factors affecting temperature measurement

The presented results were obtained under controlled laboratory conditions.

Since T_{ob} is function of input quantities that are random variables of given frequency distribution (see on expression 2.47) the uncertainty of the object temperature measurement is a model of spread of the output quantity realized around the expected value of this quantity. By sampling the

independent variables from 2.47, and assuming a certain level of correlation between them [15] simulates the resulting distribution of T_{ob}

Starting with the influence of the object's standard uncertainty of **emissivity** on the uncertainty of T_{ob}, the procedure was done considering various temperatures for the object, and by doing the simulation for various $\varepsilon_{\mathrm{ob}}$. The experiment was done for two different distances. The conclusion is that the uncertainty on the measurement of T_{ob} strongly depends on the object's temperature, *i.e.* for a given $\varepsilon_{\mathrm{ob}}$ the higher the object's temperature the higher the uncertainty will be. Besides, the assumed object's emissivity and the distance do not affect the uncertainty significantly.

Evaluating the influence of the component of the standard uncertainty of **ambient temperature**, the simulation was done by varying the object's emissivity and temperature, and was made for two different distances. The results show that there is a strong dependency on the emissivity of the object: as the emissivity decreases, the uncertainty on the object's temperature measurement increases. There is also a dependency on the object's temperature: the results show that the higher the object's temperature, the weaker the influence of the ambiente temperature uncertainty on the combined standard uncertainty. Furthermore, the influence of the ambient temperature on the temperature measurement accuracy can be neglected for very high temperatures (360 K), mainly if the object's emissivity is very high.

For the analysis of the influence of the component of the standard uncertainty of **atmospheric temperature**, the simulation was done in similar conditions as for the **ambient temperature**. The experiment led to conclude that the objtect's emissivity $\varepsilon_{\mathrm{ob}}$ does not affect the standard uncertainty of the atmospheric temperature. On the other hand, the object's temperature influences the component uncertainty associated with atmospheric temperature. The results also indicates that this influence depends on the distance, since there are different outcomes on both experiments.

When evaluating the component associated with object-camera distance, the results show that this component does not depend on the object's emissivity an it depends on the object's temperature: for a certain value of uncertainty of the distance, for a larger distance, the uncertainty on the object's temperature T_{ob} increases.

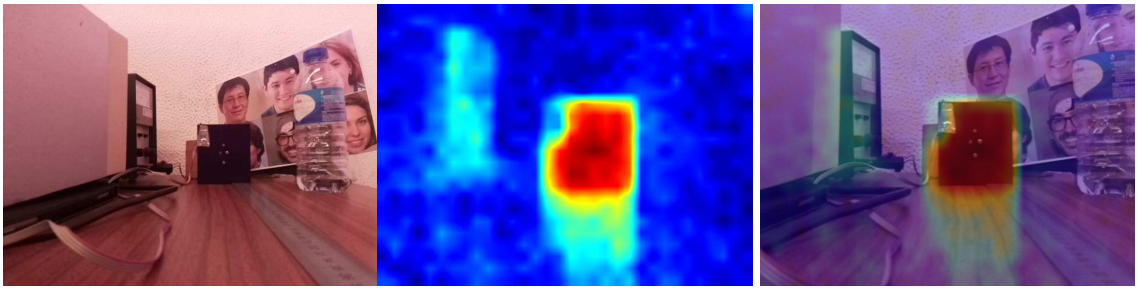
When testing the system on real faces, there can be an influence on the measurement based on the subject's skin color [32].

4.2 Determination of the homography matrix

The homography matrix, as described in Section 2.4, is obtained by combining four points on the two optical and thermal images. For that purpose the OpenCV function *cv.findHomography()*

and *cv.warpPerspective()* were used. Since this matrix is independent of the observed plane, it only takes one pair of images to perform this calibration process, as mentioned. Although it is independent from the observed plane, the result matrix will depend on which acquisition conditions it is made being only valid under the conditions that it was acquired. This will be discussed later on section 4.2.1.

In order to obtain the system's homography matrix reqired for the two-camera system calibration, two images of the an heatsink (see Section 3.2) from each camera were acquired (see Figure 4.9). Four correspondent points from each image were manually chosen using the *opencv_annotation* tool. For the sake of being easy to identify, the four corners of the heatsink were chosen.



(a) (b) (c)

Figure 4.9: The images resulting from the homography process. (a) shows the acquired optical image, while (b) shows the IR image. (c) shows the overlaying of the two images (a) and (b) after applying the homography matrix

This transformation was computed considering the object at a distance of 50 cm with the object in the center of the optical image. The homography matrix is then given by:

$$H = \begin{bmatrix} 1.395 & 0 & -165.047 \\ 0 & 1.350 & -97.651 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

4.2.1 Limitations

Doing perspective transformation brings some issues related to the accuracy of the image superposition when varying the distance between the target and the cameras. This disparity can be rectified with a triangulation method [7]. This method was not applied in this work.

4.3 Object-camera distance estimation

In our system there is no independent depth information in order to estimate the object-camera distance. Assuming that our object is containned within a plane, we can estimate its distance when applying an object detection algorithm, assuming some characteristic dimensions of the object are reasonably known. Having a bounding box surrounding the object, the distance can be estimated by relating to the box's diagonal pixel length, following the equation 4.6 [1], where $\hat{d}$ is the estimated distance, s is the diagonal pixel length, and a, h and k are constants from the fit. For our laboratory conditions, we used a six-faces scene (Figure 4.10) [6], where a Haar-cascade classifier (2.6.1) was computed to detect the faces and took s as the average diagonal length of the six faces.

Figure 4.10: Six-faces scene - the face on the middle bottom position has a hole on the forehead position, where the heatsink will be placed, in order to simulate a high temperature for the forehead.

$$\hat{d} = \frac{a}{s-h} + k \tag{4.6}$$

The plot of the fit is represented in the figure 4.11

The values for the constants were $a = 8880$, $h = -4.325$, and $k = -3.7$, with an $R^2 = 0.9988$. From these values, the equation 4.6 was introduced to the code in order to automatically estimate the object-camera distance.

The same estimation was made for real faces[7], having the constants $a = 12940$, $h = 33.84$, and $k = 4.23$, with an $R^2 = 0.9446$. This result will be used later on section 4.4.2

[6] Since the shown people does not exist, *i.e.* the faces were randomly generated from an algorithm [33], their faces were not blurred

[7] Note that the expression 4.6 depends on the apparent size captured by the camera. Consequently, since the real faces have different real sizes than the printed faces, a new fit must be performed for the real faces.

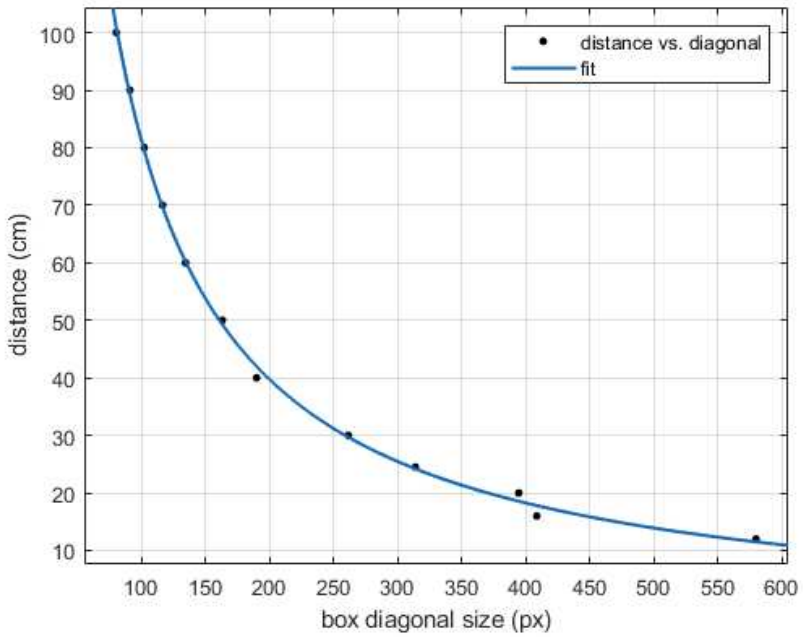


Figure 4.11: Fit for distance estimation

4.4 Haar-cascade algorithm performance

By combining the optical and the IR images using the established homography matrix given by 4.5, the algorithm described on section 2.6.1 was applied.

This part of the experiment served to test the temperature compensation model 2.5.2 defined and to test the overall system performance at various distances. In a first approach (section 4.4.1) we tested the algorithm using the six-face scene (Figure 4.10) and its automatic distance estimation through the equation 4.6. Later, in comparison, the system was tested with real faces, with a different set of parameters for the fit on equation 4.6.

In both experiments we searched for the subjects' forehead and monitored their temperature. In the context of face detecting classifiers, OpenCV only provides already trained classifiers for the frontal view of face, eyes, and eyes with glasses. Since there isn't a direct way to estimate the forehead position, the estimation was made, considering the coordinates of the face and the eyes (through the upper limit of the face box and the coordinates corresponding to the upper left corner of the left eye and the upper right corner of the right eye), as shown on figure 4.12. From this estimation, the corresponding coordinates are matched in the IR image, being then able to present the forehead's temperature. The forehead estimation is limited by the detection of the eyes. It is only able to estimate when there are two well-estimated eyes. Our system is also designed to show the temperature on the subject's forehead (in Celsius degrees), when detected.

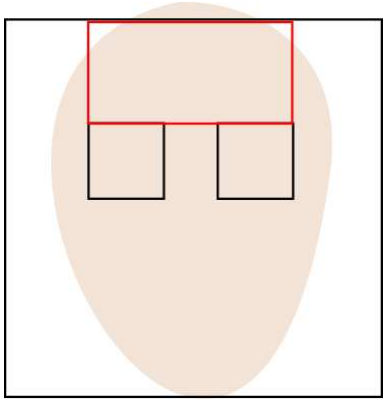


Figure 4.12: Example of forehead coordinates estimation - the oval shape observed is an example of a face. The larger box that surrounds the face is the square that comes from the face detetection of the Haar-casacde algorithm, where the eyes are also detected and represented by the two smallest black squares. The forehead position (red rectangle) is then estimated from the top corner of the left eye, the top right corner of the right eye, and the upper limit of the bounding box surrounding the face.

4.4.1 Testing on face simulation

This simulation was done having a six-scene face with one simulated hot forehead from an heated heatsink in thermal equilibrium, as seen on figure 4.10. The temperature of the heatsink was monitored with a TPMx75 sensor. The simulation was done considering the defined optimal distance operation range, and the results are shown from figure 4.15 to 4.18.

(a) (b)

Figure 4.13: Setup for face simulation - the hole on the bottom middle subject is used to simulate a high temperature forehead.

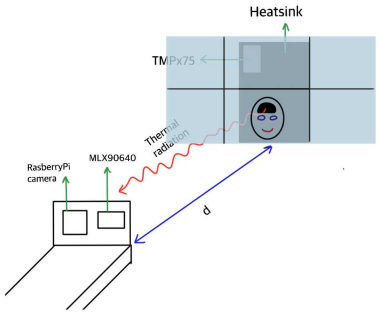
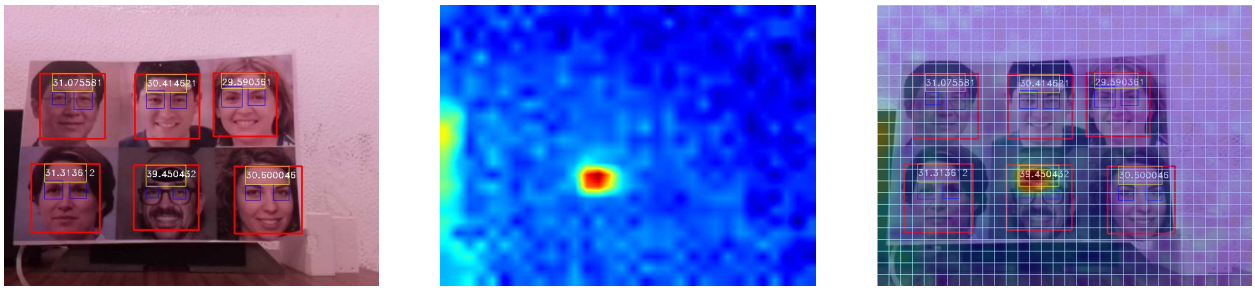
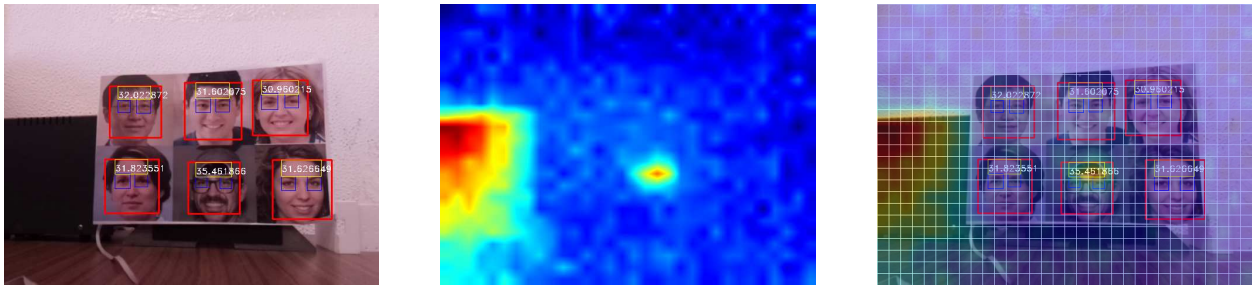


Figure 4.14: Schemactic representation for the experimental setup. Since the heatsink is at high temperature, the camera will acquired the high temperature correspondent to the hole, assuming it is the subject's forehead temperature



(a) (b) (c)

Figure 4.15: Obtained results for 50 cm. (a) shows the performance of the Haar-cascade algorithm; (b) shows the thermal image from MLX90640; (c) shows an overlay of (a) and (b)

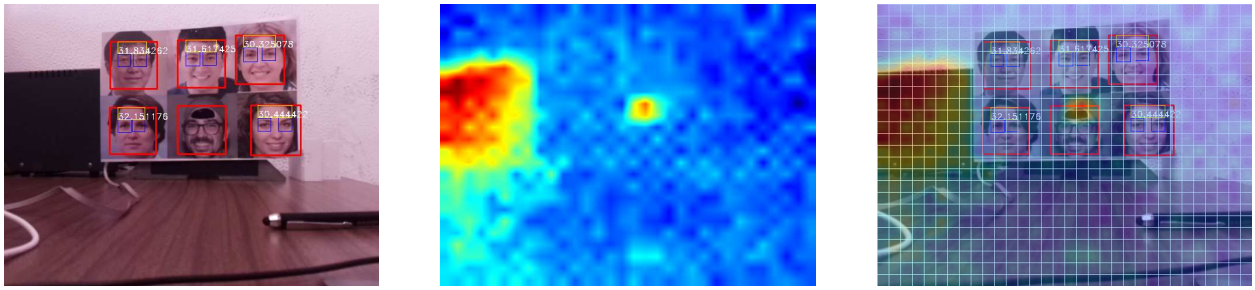


(a) (b) (c)

Figure 4.16: Obtained results for 60 cm. (a) shows the performance of the Haar-cascade algorithm; (b) shows the thermal image from MLX90640; (c) shows an overlay of (a) and (b)

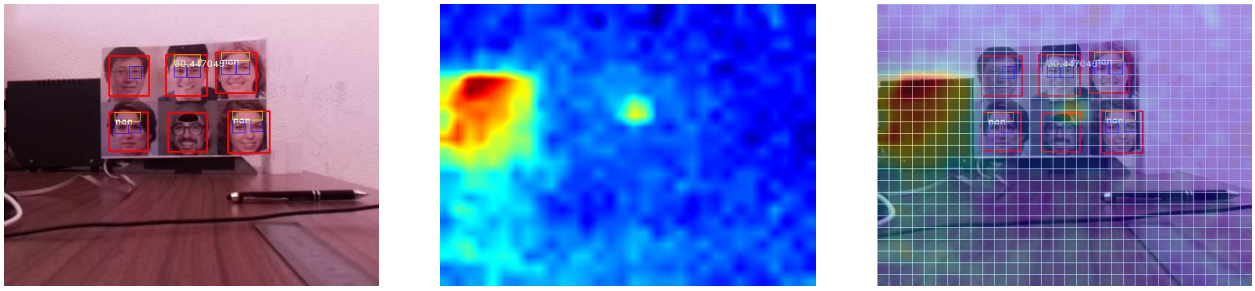
When looking in the classifier performance, it does not work very well at larger distances (70cm and 80cm) [8]. Even when looking to the forehead position estimation, the system is limited, not being able to detect a forehead when the height is less or equal to a single row, column or

[8]Since these faces are printed images, the algorithm performance also depends on the size of the printed images



(a) (b) (c)

Figure 4.17: Obtained results for 70 cm. (a) shows the performance of the Haar-cascade algorithm; (b) shows the thermal image from MLX90640; (c) shows an overlay of (a) and (b)



(a) (b) (c)

Figure 4.18: Obtained results for 80 cm. (a) shows the performance of the Haar-cascade algorithm; (b) shows the thermal image from MLX90640; (c) shows an overlay of (a) and (b)

pixel of the thermal matrix. There is also other limitation on this approach. Since the hot subject's forehead looks black, due to the color of the heatsink, the classifier has some issues detecting the face and the eyes. This can be improved by working with real faces (see section 4.4.2) instead of the printed ones.

Knowing the equilibrium temperature of the heatsink (∼50ºC), the results of the temperature simulation seem too low. When looking on the IR matrix, there is some clear high-temperature pixels, but the contrast between the paper temperature (∼30ºC) and the heatsink is high enough to introduce thermal noise to the measurement. It is then expected that the measured temperature is lower than the real heatsink temperature, reducing the measurements' accuracy. When applying this algorithm into real faces it would be expected this issue not to occur since the subject's temperature will be more homogeneous. Therefore, the second part of this experiment will be done on real humans.

4.4.2 Testing on real faces

Experiment at 32Hz refresh rate

This part of the experiment was made with a set refresh rate of 32Hz. The temperature of the human body was measured with a regular digital thermometer [9] and its temperature was $T_{body} = 35.6°C$. The room temperature was $T_{room} = 25°C$. The human's skin emissivity was considered $\varepsilon = 0.98$ [32]. Figures 4.19 to 4.21 show the imaging results of the experiment. Table 4.3 shows the results for the experiment, following the steps of the thermal compensation, already defined on section 2.5.2. The compensated temperature is expressed as $\hat{T}$.

Distance (m)	Digital thermometer (ºC)	MLX90640 (ºC)	$\hat{T}$
0,5	35,6	35,0	35,5
0,6	35,6	35,1	35,9
0,7	35,6	34,6	35,6
0,8	35,6	34,0	35,3
0,9	35,6	34,0	35,6
1,0	35,6	34,8	35,7

Table 4.3: Results for the temperature estimation for the acquisition

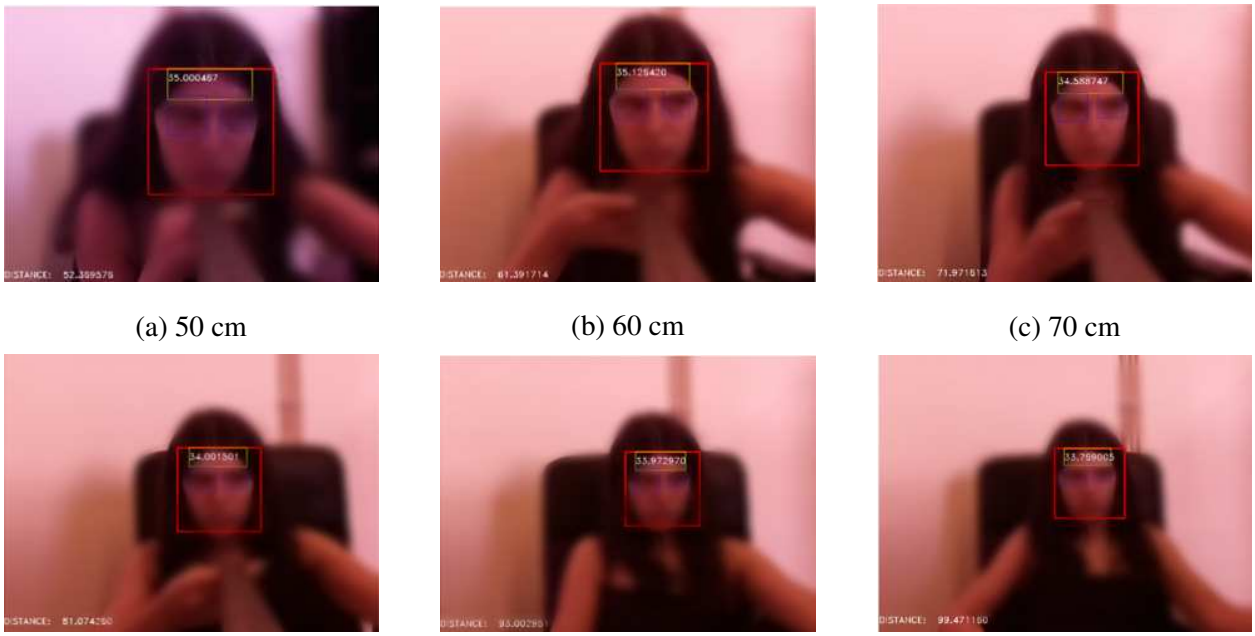
Table 4.4 shows the MAE (eq. 4.4) before and after applying the temperature compensation model.

	MAE
Before compensation	1.0
After compensation	0.1

Table 4.4: Mean absolute error before and after temperature compensation

At closer distances (0.5-0.6 m) the acquired temperature by MLX90640 is still very close to the one measured with a digital thermometer, and the absolute error increases as the subject moves away from the camera. The results show that the MAE decreases from 1.0 to 0.1 after the compensation, shown effectively that the compensation model improves the system's accuracy.

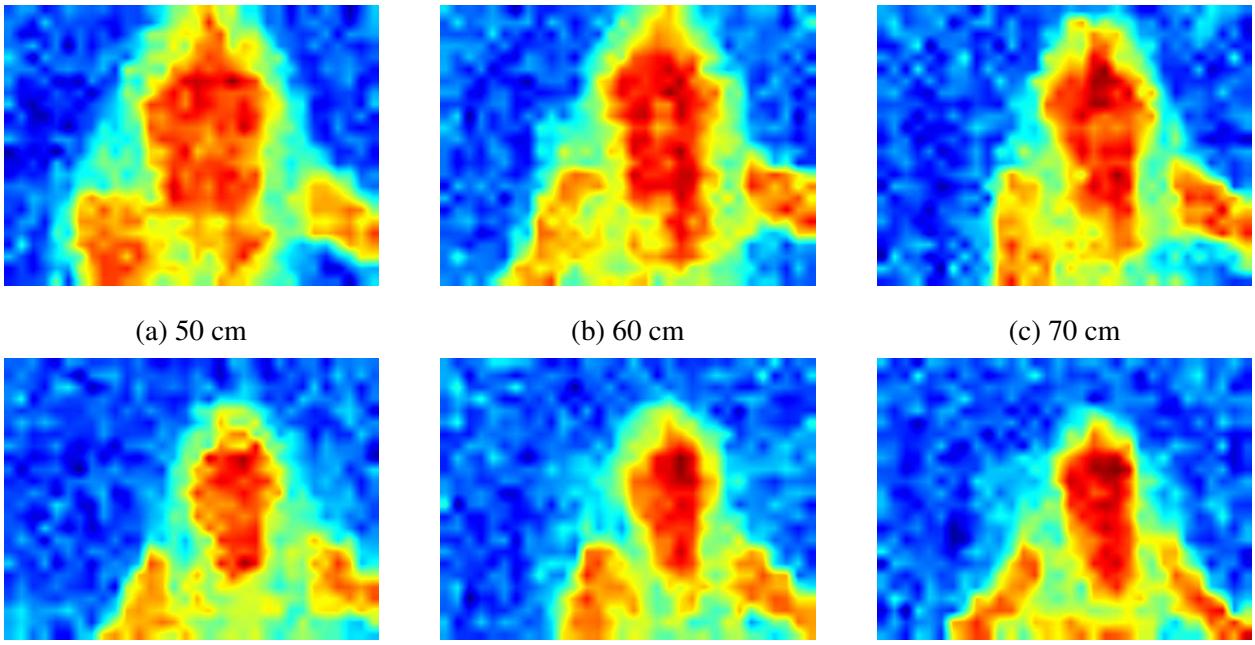
[9] The used digital thermometer was the one commonly used to measure fever at home, for example, placed in the armpit. Its uncertainty is 0.1°C



(a) 50 cm (b) 60 cm (c) 70 cm

(d) 80 cm (e) 90 cm (f) 100 cm

Figure 4.19: Optical acquisition on a real face, at various distances



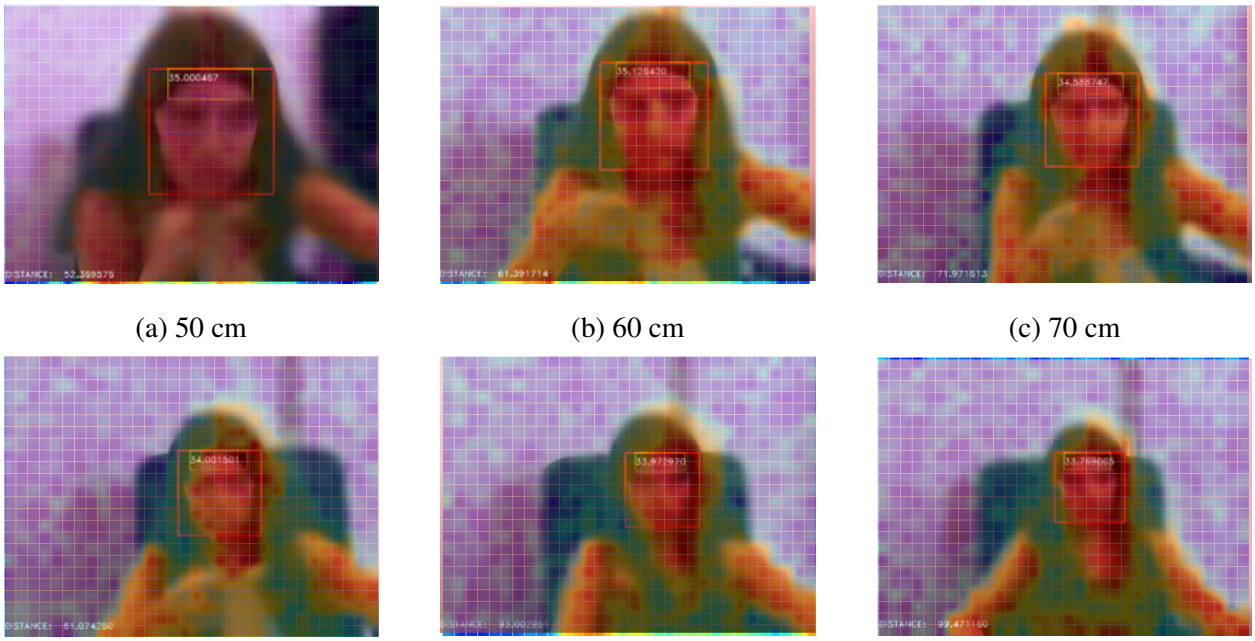
(a) 50 cm (b) 60 cm (c) 70 cm

(d) 80 cm (e) 90 cm (f) 100 cm

Figure 4.20: Thermal acquisition on a real face, at various distances

Experiment at 8Hz refresh rate

In order to verify if the refresh rate would influence on our results, since we already showed that it impacts the noise RMS, this part of the experiment was made with a set refresh rate of 8Hz. The temperature of the human body was measured with a regular digital thermometer and



(a) 50 cm (b) 60 cm (c) 70 cm

(d) 80 cm (e) 90 cm (f) 100 cm

Figure 4.21: Overlay of the the optical and thermal acquisition on a real face, at various distances

its temperature was $T_{body} = 35.3°C$. The room temperature was $T_{room} = 25°C$. The human's skin emissivity was also considered $\varepsilon = 0.98$ [32]. Figures 4.22 to 4.24 show the imaging results of the experiment. Table 4.5 shows the results for the experiment, following the steps of the thermal compensation, already defined on section 2.5.2. The compensated temperature is expressed as $\hat{T}$.

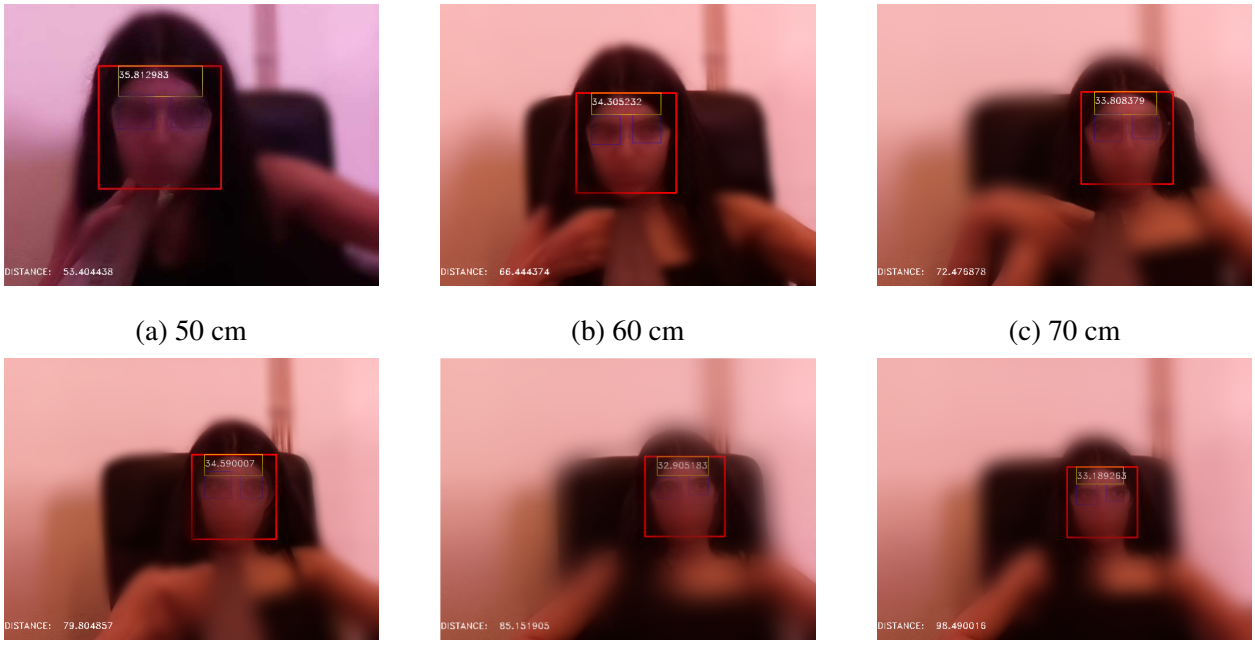
Distance (m)	Digital thermometer (ºC)	MLX90640 (ºC)	$\hat{T}$
0,5	35,3	35,8	36.4
0,6	35,3	34.3	35.2
0,7	35,3	33,8	35,0
0,8	35,3	34,6	36.2
0,9	35,3	32,9	34.7
1,0	35,3	33,2	35,4

Table 4.5: Results for the temperature estimation for the acquisition - Refresh rate 8Hz

	MAE
Before compensation	1.5
After compensation	0.5

Table 4.6: Mean absolute error before and after temperature compensation - Refresh rate 8Hz

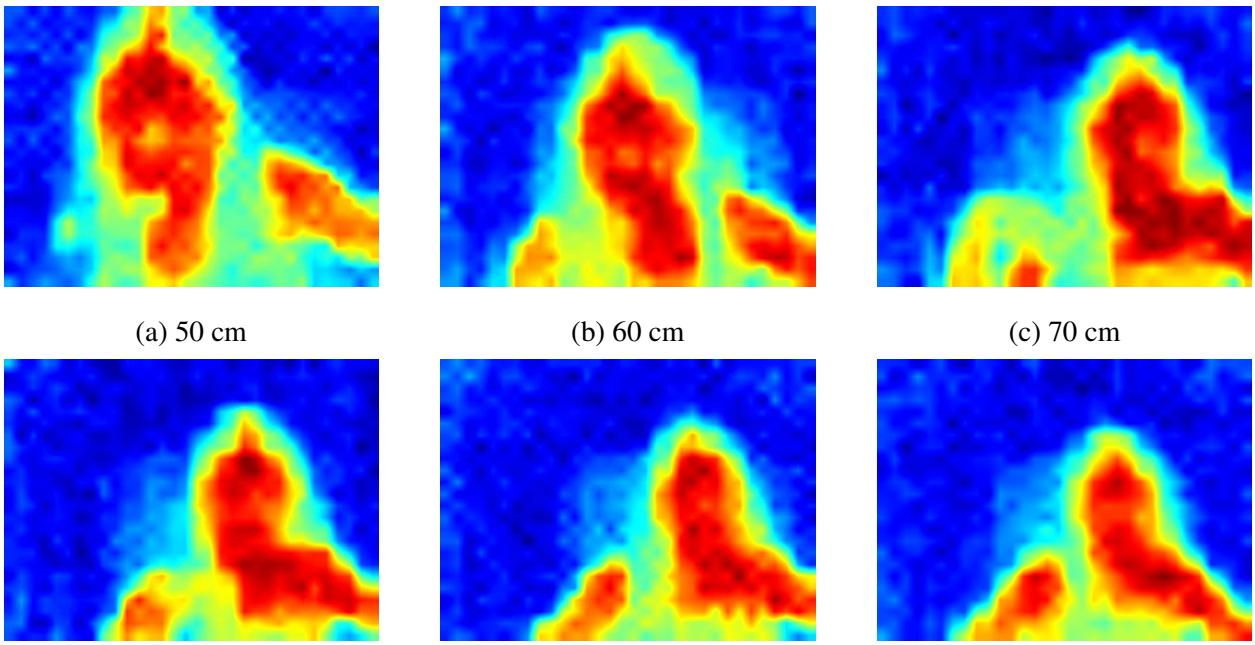
Table 4.6 shows the MAE (eq. 4.4) before and after applying the temperature compensation model. When looking to the absolute values of the temperature acquired by the MLX90640 camera, although the MAE decreases after applying the compensation model, the 32Hz refresh rate gives a lower MAE on the overall measurements, when compared to the 8Hz refresh rate. This is unexpected since there abnormal fluctuations in the temperature measurement, which end up influencing the obtained results. However this may be linked with issues in the acquisition software, such as the set refresh rate - the statistical fluctuations in each acquisition become more evident when acquiring with a lower set refresh rate.



(a) 50 cm (b) 60 cm (c) 70 cm

(d) 80 cm (e) 90 cm (f) 100 cm

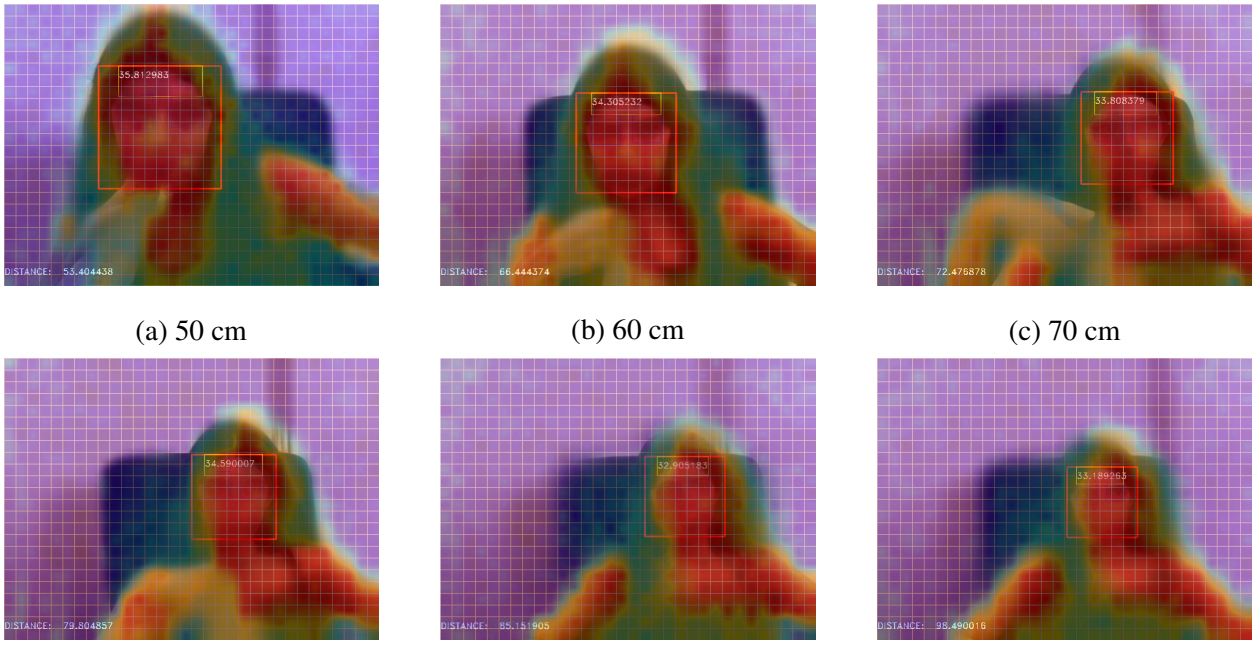
Figure 4.22: Optical acquisition for a set refresh rate = 8Hz on a real face, at various distances



(a) 50 cm (b) 60 cm (c) 70 cm

(d) 80 cm (e) 90 cm (f) 100 cm

Figure 4.23: Thermal acquisition for a set refresh rate = 8Hz on a real face, through various distances



(a) 50 cm (b) 60 cm (c) 70 cm

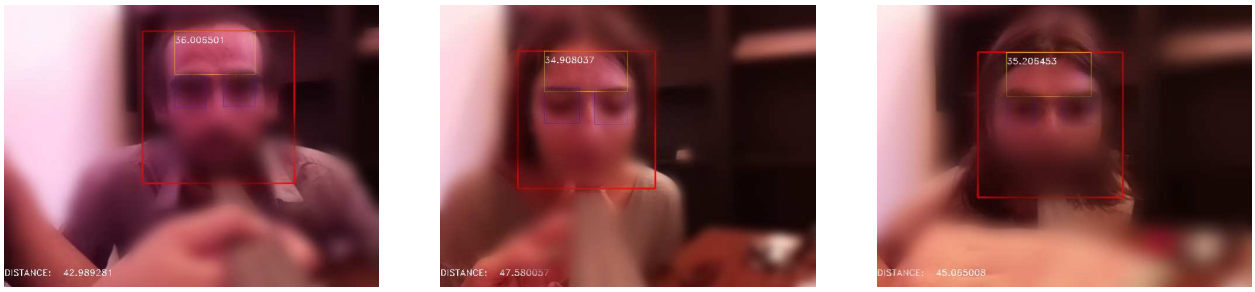
(d) 80 cm (e) 90 cm (f) 100 cm

Figure 4.24: Overlay of the the optical and thermal acquisition for a set refresh rate = 8Hz on a real face, through various distances

4.4.3 Validating the system with three different people

In order to validate the results with different persons/faces, a small set of three people was chosen. The experiment was made at a controlled distance of $d = 50cm$, with $T_{room} = 25°C$.

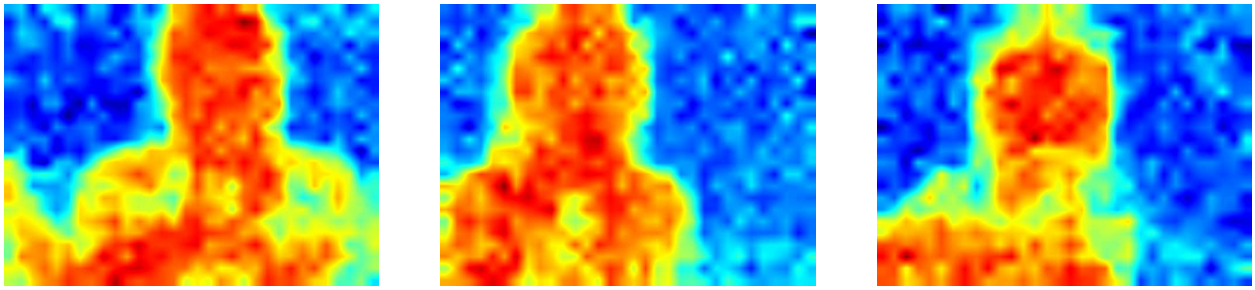
Figures 4.25 and 4.26 show the optical an thermal acquisitions, respectively. Figure 4.27 shows the overlay of the thermal and optical images, with a grid to help the visualization of the 24x32 matrix. Table 4.7 shows the comparison between the measurements and the control temperature, acquired with a regular digital thermometer[10].



(a) Subject 1 (b) Subject 2 (c) Subject 3

Figure 4.25: Validation set using three different people - optical acquisition

[10]The used digital thermometer was the one commonly used to measure fever at home, for example, placed in the armpit. Its uncertainty is 0.1°C



(a) Subject 1 (b) Subject 2 (c) Subject 3

Figure 4.26: Validation set using three different people - thermal acquisition

(a) Subject 1 (b) Subject 2 (c) Subject 3

Figure 4.27: Validation set using three different people - overlay of the acquisitions (thermal and optical)

Subject	Thermometer (ºC)	System (ºC)	Absolute difference	Noise RMS (forehead) (ºC)
1	35.8	36.0	0.2	0.6
2	36.9	34.9	2.0	0.8
3	36.5	35.2	1.3	0.9

Table 4.7: Comparison between the temperature acquired by a digital thermometer and by the system. The absolute error is very different for each subject and the noise RMS is also changing.

This experiment was performed at the same distance that the homography matrix was calculated. When evaluating the correspondence of the optical and thermal images through homography, we see that these images are perfectly matched (see the overlay of the two images on figure 4.27). This leads to the conclusion that the matrix is still adequate.

From table 4.7 we see that there is a huge difference on the absolute error between Subject 1 and the remain subjects. Besides that, the noise RMS does not stay constant and for Subject 1 and 2 it is different from the noise RMS for a refresh rate of $32Hz$ (see table 4.1). This suggests that there is other parameters influencing the forehead's temperature. When looking on the figures of the overlay (Figure 4.27) and the thermal acquisition 4.26, we can see that there is a influence of

the subject's hair on the measurement. When comparing with Subject 1, Subjects 2 and 3 have much more quantity of hair on the estimated region. This leads to acquiring lower temperatures (and higher noise RMS), due to the presence of hair on the forehead.

The distance estimation is not very accurate. The distance fit is based on the diagonal size of bounding box detecting the face. The fit made for real faces, defined on section 4.3, was made with a specific subject's face, under specific dimensions. It is expected, since we are dealing with randomly picked subjects that the fit's accuracy decreases, and the faces have different sizes and shapes.

4.5 Overall system's performance and limitations

In this section we will sum up the operating conditions of our system, evaluated and defined through our analysis. Since the system must always be supervised, the supervisor is responsible for assure that some errors or limitations doesn't occur. When testing the system and its algorithm, there was some reported limitations on the system.

4.5.1 MLX90640 operating conditions

An evaluation of the influence of the refresh rate on the noise RMS was made and took the conclusion that the higher the refresh rate, the higher the noise RMS (see table 4.1)

Due to our thermal acquisition analysis, an optimal distance operation range was defined. It is considered as $50cm$ to $100cm$ (see Section 4.1.6).

Since cameras have different resolution, FOV, and are separated by a certain distance, there is the need of calibrating both acquired images, turning the observed images into a merged single observed plane. Since the optimal distance operation range begins at $d = 50cm$, the homography matrix was calculated for that distance. Because of being determined in such distance, when measuring at different distances, there will be a deviation on the overlay of the two images (approximately 2 pixel to the right for 100cm - due to the relative position of the cameras - see on Figures 4.17, 4.18, 4.21, and 4.24).

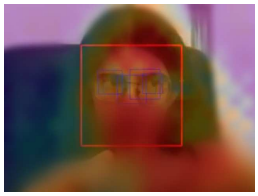
Besides that, the system is limited by the influence of surroundings (and that influence depends on the distance). In order to understand this behaviour, the system must be pre-calibrated following the steps of 2.5.2, for each object under measurement.

4.5.2 Other limitations

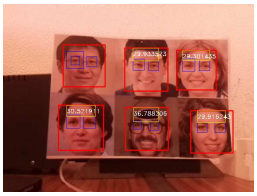
The system must always have a supervisor to ensure the system's correct operation. Through this, some issues with the system and the algorithm were detected. This section mentions the detected problems to prepare the supervisor for them.

Beginning with **feature detection**, since the images may contain complex sources of various features, this can induce erroneous detection of faces and eyes, using Haar-cascade classifiers. For instance, when the face of the subject under analysis is close to the system, the nostrils can induce eye detection due to their features, which the algorithm considers to be eyes. Faces with eyeglasses can make the task of detecting eyes more difficult. On figure 4.28 there are some examples of wrong feature detection.

(a) Error of face detection



(b) Error on glasses on real face



(c) Error on glasses on simulation

(d) Error on nostrils for eye detection

Figure 4.28: Example of errors on the algorithm detection

Our system is oriented to estimate the position of the foreheads, from the coordinates of the detected face and eyes. The forehead may not be an homogeneous surface, due to the presence of hair. This can lead to measure lower temperatures than expected. Besides that, other errors can be introduced for the optical corresponding thermal pixel estimation. The position is estimated by a quotient between the coordinates of the two images and corresponding matrices. This leads to bigger estimated areas, which can lead to more inaccurate measurements.

As mentioned, the system must be supervised. It is the supervisor's responsibility to ensure that the **emissivity** of the object under study is defined in the code of our system, and must be changed everytime the object changes.

4.5.3 Requirement verification

In order to verify if our system meets the requirements defined on Section 1.2, we tested some parameters during our analysis.

For a set refresh rate of 32Hz, we calculated the time that took to perform the algorithm loop, for 50 loops. The average time of the loop took was 0.000839 seconds, that corresponds to a acqui-

sition rate of $\sim$1192 Hz, which checks the system's requirements (analysis rate <1Hz). Although RaspberryPi is a system with relatively rescricted capabilities it can be used for a final system, and there is a possibility for using more robust detection algorithms, for face, eyes and forehead detection.

The analysis presented in the sections 4.1.7, 4.4.2, and 4.4.3) at the optimal distance operation range, and without the application of the temperature compensation model, 53% of the measurements check the requirement for the measurement precision being <1ºC. When applying the temperature compensation model (in sections 4.1.7 and 4.4.2), all of the results comply this requirement, except the ones from table 4.7. Therefore, the compensation model ensures that this requirement is met.

5 Conclusions

A system for temperature measurement at a object-camera distance range of 50-100cm (Section 4.1.6) using a combination of optical and thermal cameras was developed, that can measure a subject's forehead temperature. In this study, the system's properties and its operation mode were evaluated and explained throughout the discussion.

The system analysis began by studying the operating properties of the IR camera (MLX90640). To understand the stabilization time of the camera before a measurement, the study of this property was performed, and led to the conclusion that the stabilization time was much shorter than the acquisition period and can be neglected. Since not all the auto calibration details of the MLX90640 camera are disclosed by the manufacturer, an analysis was performed to try to understand the conditions that were assumed for that calibration in terms of the parameters of the theoretical model, suggested by [15], Right away, the relation between the noise RMS and the camera's refresh rate was evaluated, and compared with the manufacturer data. The results shown that the RMS noise increase with the refresh rate, agreeing with what is supplied in the datasheet.

For the temperature measurement analysis, the analysis started by evaluating an heatsink's temperature, in thermal equilibrium, with known properties. The temperature was measured with an MLX90640 camera and then compared to the readout of a TMPx75 sensor. The results showed that there was a strong dependency of the object-camera distance on the measured temperature. The noise RMS on the considered RoI were consistent with the RMS noise previously calculated for the chosen refresh rate of 32Hz. The maximum deviations on this measurements were evaluated. The results on the maximum deviation on a sliding window around the manual selection by one pixel show that the deviation increases as the object moves away from the camera, due to the edge effect error. Analysing the evaluation of the temperature measurement through various distances, the conclusion was that at closer distance there was a bias on the measurements, which is thought to be related with the conditions of the cameras auto calibration. From that analysis an optimal distance operation range was defined as 50cm to 100cm.

To improve the measurements, the compensation model presented on Section 2.5.2 was tested,

for the optimal distance operation range. The MAE after the compensation decrease from 0.7 to 0.2, which led to conclude that the model improved the system's accuracy on temperature measurement. The model can be improved by considering other fit models beyond linear.

Once the thermal analysis was completed, it was time to merge the functionalities of the thermal and optical cameras, through homography, in order to perform the imaging calibration. However, this determination was limited for much greater distances than that at which the homography matrix was determined. Since the cameras have two different FOV and different positioning on the enclosure, the accuracy when overlaying the images decrease as the distance increases. However, this does not represent a limitation for the system performance, since at larger distances the performance of IR camera decreases, as the subject will only fulfill a few pixels.

For distance estimation, and since there was no independent depth information on the system, a model was assumed which compares the real object-camera distance to the bounding box obtained through an Haar-cascade classifier. The results would then be used in a simulation for forehead's temperature estimation and for a real face experiment.

Since the Haar-cascade algorithm provided by OpenCV did not have a pre-trained classifier for foreheads, the estimation of the forehead's position was estimated in a indirect way, from the head and eyes positioning. The test for this algorithm was made with with a simulation with non-real faces, with a heated forehead simulated by an heated heatsink. The results were not very good since the measured temperature was much lower than the real heatsink's temperature, due to the contrast between the scene and the heatsink's surface. For real face temperature measurement, the experiment was done at two different refresh rates (32Hz and 8Hz). The results on the algorithm performance improved, since it worked for all operating distances, since the mentioned issue detecting the forehead from the previous analysis no longer existed. Applying the thermal compensation model, the MAE of the measurements decreased, from 1.0 to 0.1, which mean that the system's accuracy increased by applying the compensation model. For a 8Hz refresh rate, the experiment was repeated under the same conditions, but the results were worse (MAE decreased from 1.5 to 0.5) - this is thought to be due to unrelated issues with the acquisition software. For both experiments, as the distance increased, there was a little deviation on the matching images, due to the limitation of the homography process, previously mentioned.

The system was then validated with a set of three different subjects, by measuring their forehead's temperature at the optimal distance of 50 cm. The results were satisfying, but there was differences on the absolute error of the subjects, probably because of the presence of the hair on the measurement, which were different for each subjects.

Finally, summing up the overall system's performance and its limitations, the operation condi-

tions were defined, such as the distance operation range, the influence of the RMS, the deviation on the overlaying two images as the distance increases and its calibration for distance-measuring purposes. There was also mentioned other limitations such as the erroneous feature detection, the image update on the user interface, the likely erroneous estimation of the forehead due to the calculation estimation, the distance-fit calibration, and the object's emissivity change everytime the object's under measurement is changed.

5.1 Future work

In this section some suggestions are proposed in order to improve the system's performance.

When dealing with an object with known and well defined properties, the system can be optimized on detecting automatically a compensated temperature, from a pre-evaluation of the system's behaviour. If the influence of the subject-camera distance is known, the fit of the factor y can be introduced in the algorithm and being estimated for every interest distance. This will able to know the compensated temperature at real time operation. However, this evaluation is limited for other objects or subjects, being necessary a different evaluation for each object.

The automatic distance estimation can be also be improved, by using for instance two optical cameras, or add to the system a laser capable of measuring the distance.

The ML component on this work can be expanded with deep learning models for CV. The used algorithm is limited due to its properties, since it is less accurate than the techniques based on deep learning. Besides, the train task on a complex object can be very difficult, and it is done manually. This can lead to a high false-positive rate.

Furthermore, in order to improve the system's performance and reducing the deviation on the overlaying effect, the triangulation effects must be studied and applied in the system. Knowing this behaviour, the distance effect on the overalying image will decrease and therefore the accuracy will increase.

Bibliography

[1] Jing Wei Chin, Kwan Long Wong, Tsz Tai Chan, Kristian Suhartono, and Richard HY So. An infrared thermography model enabling remote body temperature screening up to 10 meters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3875–3881, 2021.

[2] Rikke Gade and Thomas B Moeslund. Thermal cameras and applications: a survey. *Machine vision and applications*, 25(1):245–262, 2014.

[3] Begoña C Arrue, Aníbal Ollero, and JR Matinez De Dios. An intelligent system for false alarm reduction in infrared forest-fire detection. *IEEE Intelligent Systems and their Applications*, 15(3):64–73, 2000.

[4] Diego A Socolinsky and Andrea Selinger. Thermal face recognition over time. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 187–190. IEEE, 2004.

[5] Socheat Sonn, Guillaume-Alexandre Bilodeau, and Philippe Galinier. Fast and accurate registration of visible and infrared videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 308–313, 2013.

[6] Surya Prakash, Pei Yean Lee, Terry Caelli, and Tim Raupach. Robust thermal camera calibration and 3d mapping of object surface temperatures. In *Thermosense XXVIII*, volume 6205, pages 182–189. SPIE, 2006.

[7] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[8] Duane C Brown. Decentering distortion of lenses. *Photogrammetric Engineering and Remote Sensing*, 1966.

[9] AE Conrady. Lens-systems, decentered. *Monthly notices of the royal astronomical society*, 79:384–390, 1919.

[10] Basic concepts of the homography explained with code = https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html, note = Accessed: 2022-06-22.

[11] Lecture 16 - planar homographies. https://www.cse.psu.edu/~rtc12/CSE486/lecture16.pdf. Accessed: 2022-08-24.

[12] Opencv documentation - basic concepts of the homography explained with code. https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html. Accessed: 2022-08-24.

[13] Homography estimation = https://towardsdatascience.com/estimating-a-homography-matrix-522c70ec4b2c, note = Accessed: 2022-06-25.

[14] Giovanni Battista Dell'Isola, Elena Cosentini, Laura Canale, Giorgio Ficco, and Marco Dell'Isola. Noncontact body temperature measurement: uncertainty evaluation and screening decision rule to prevent the spread of covid-19. *Sensors*, 21(2):346, 2021.

[15] Waldemar Minkina and Sebastian Dudzik. *Infrared thermography: errors and uncertainties*. John Wiley & Sons, 2009.

[16] What are swir, mwir, and lwir – and what do they mean? = https://www.dronegenuity.com/what-are-swir-mwir-and-lwir-and-what-do-they-mean/, note = Accessed: july 2022.

[17] Jing Chen, Ji-Ping Wang, Tian-Yu Shen, Da-Xi Xiong, and Li-Quan Guo. High precision infrared temperature measurement system based on distance compensation. In *ITM Web of Conferences*, volume 12, page 03021. EDP Sciences, 2017.

[18] Yu-cun Zhang, Yi-ming Chen, Xian-bin Fu, and Cheng Luo. A method for reducing the influence of measuring distance on infrared thermal imager temperature measurement accuracy. *Applied Thermal Engineering*, 100:1095–1101, 2016.

[19] Tmpx75 datasheet. https://www.ti.com/lit/gpn/tmp75. Accessed: 2022-08-26.

[20] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001.

[21] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[22] Poorvi Taunk, Vani Jayasri Geddada, Padma Priya J, and N. Suresh Kumar. Face detection using viola jones with haar cascade. *Test Engineering and Management*, 83, 06 2020.

[23] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International journal of computer vision*, 38(1):15–33, 2000.

[24] Cascade classifier training = https://docs.opencv.org/3.4/dc/d88/tutorial_traincascade.htm, note = Accessed: july 2022.

[25] Raspberrypi illustration = https://commons.wikimedia.org/wiki/file:raspberry_pi_b%2b_illustration.svg, note = Accessed: march 2022.

[26] Sketchup tool = https://www.sketchup.com/products/sketchup-for-web, note = Accessed: october 2021.

[27] Rasperrypi camera module v2 = https://pt.rs-online.com/web/p/camaras-para-raspberry-pi/9132673, note = Accessed: september 2022.

[28] Rasperrypi camera documentation = https://www.raspberrypi.com/documentation/accessories/camera.html, note = Accessed: march 2022.

[29] Adafruit mlx90640 ir thermal camera. https://cdn-learn.adafruit.com/downloads/pdf/adafruit-mlx90640-ir-thermal-camera.pdf?timestamp=1581704431. Accessed: 2022-06-23.

[30] Anodized aluminum heatsinks: What you need to know. https://www.gabrian.com/anodized-aluminum-heatsinks-what-you-need-to-know/#:~:text=The%20emissivity%20of%20an%20anodized,be%20measured%20at%20around%200.85. Accessed: 2022-08-26.

[31] Mlx90640 driver = https://github.com/melexis/mlx90640-library/blob/master/mlx90640%20driver.pdf.

[32] Matthew Charlton, Sophie Stanley, Zoë Whitman, Victoria Wenn, Timothy Coats, Mark Sims, and Jonathan Thompson. The effect of constitutive pigmentation on the measured emissivity of human skin. *PLOS ONE*, 15:e0241843, 11 2020.

[33] This person does not exist - random face generator = https://thispersondoesnotexist.com/, note = Accessed: 2021.