International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-8, Issue-8, August 2022
ISSN: 2395-3470
www.ijseas.com

Temperature Detection Using an IR Camera

José-Ignacio Vega-Luna[1], Gerardo Salgado-Guzmán[1], José-Francisco Cosme-Aceves[1], Francisco-Javier Sánchez-Rangel[1], Víctor-Noé Tapia-Vargas[1]
[1]Departamento de Electrónica-Área de Sistemas Digitales, Universidad Autónoma Metropolitana-Azcapotzalco, Ciudad de México, México

Abstract
The COVID-19 pandemic has created a need to increase hygiene standards, social distancing and vital signs measurement both in homes, offices, schools, hospitals and in any public place. One of these signs is temperature. It is widely recommended to determine the temperature of people at access points, automatically and without contact, to avoid an increase in contagion and facilitate mobility. In recent years, advances in technology have led to the appearance of electronic devices, such as sensors, IR cameras and embedded modules as well open-source software that allow the development of measurement and monitoring systems in an agile and economical way. This paper presents a system for detecting the temperature of people and objects using an IR camera and a Raspberry Pi 4 computer module. The software running on the Raspberry Pi module reads the temperature value delivered by the array of 24x32 sensors of the camera, processes the information and displays the thermal images of the silhouettes of people and objects on an HD monitor. The results of the tests conducted showed that the system can display 2.4 frames per second on the monitor using moderate operating speeds and the average accuracy of the temperature measurement is 99.34% at 2 meters.
Keywords: *COVID-19, IR camera, Raspberry Pi 4, temperature detection, thermal images.*

1. Introduction

Currently, the COVID-19 pandemic has resulted in an intensified use of both health measures and devices to measure people's vital signs. One of these vital signs is body temperature, the monitoring of which helps to prevent an increase in infections. In almost all companies, institutions, shopping centers and hospitals, the temperature of people is measured using infrared thermometers, avoiding any contact. However, this procedure takes some time and is, to a certain extent, difficult, since in some place's bottlenecks, traffic congestion and crowds are generated, which can be counterproductive. A solution that is used for mass monitoring of body temperature in certain facilities is the use of a thermal imager or a thermal camera [1]. A thermal imager is a device that allows you to see the temperature of objects and people in real time without storing them. They are used for inspecting hotspots in places where there is equipment that generates heat. A thermal, or tomographic camera, stores the captured images for process them later.

The operation of cameras and thermal imagers is analogous to that of common, or standard, video cameras since both use light to capture images. The difference is that the first detect and filter light, they only recognize the infrared region of the electromagnetic spectrum and not the region of visible light [2]. The use of infrared sensors to measure temperature, without having physical contact with the body, was patented when the correspondence between radiation and the heat emitted by black bodies was discovered [3]. The use of cameras and thermal imagers is a measure that has generated positive results, since they produce images with reliable results, however, they are used in few places because they are not affordable. Depending on the type and functionalities of the camera, or imager, the price is from hundreds to thousands of dollars, or the rent is from tens of dollars per day [4]. These devices provide images like the one shown in Figure 1.

However, technological advances in recent years have made it possible to create integrated circuits and IR detectors used in various applications, such as the detection of bodies through the heat emitted for non-invasive medical treatments and tests. One of the benefits of the above is the possibility of obtaining thermal images using low-cost IR cameras composed of arrays of infrared (IR) sensors. Until a few years ago, the resolution of

these devices was relatively low, 8x8 sensors, however, today it is possible to purchase cameras of this type with higher resolution and much cheaper than a thermal camera or imager. Using the heat emitted by the human body, the temperature pattern and the silhouette of people can be determined [5]. An IR camera made up of an inexpensive, compact size IR sensor array that has become one of the most popular in various applications of this type is the MLX90640 IR camera whose resolution is 24x32.

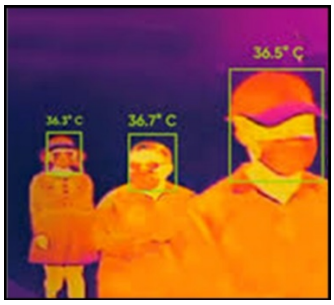

Fig. 1. Image obtained with a thermal camera.

The objective of this work was to develop a body temperature detector using a MLX90640 camera integrated by an array of IR sensors. The system works based on a computer module, or embedded module, Raspberry Pi, which is in charge of continuously reading the digital output delivered by the IR camera. The image of the silhouettes of the people located in the angle of coverage of the camera is shown on an HD monitor, indicating, through distinct colors, the temperature emitted by the human body and objects, similar to what is shown by a thermal camera, or imager. The purpose of using the IR camera is to visually indicate, without any contact, the temperature of people and objects, to prevent and avoid situations of health risk.

Reviewing the state of the art regarding the research conducted in recent years with IR sensors, the work has focused on the development of high sensitivity microelectromechanical systems (MEMS) that incorporate amplification and signal processing using chopper operational amplifiers [6] or silicon-on-insulator (SOI) based membrane technology sensors to reduce pixel crosstalk [7]. In parallel, given that obtaining infrared thermal images is a non-invasive, fast process that covers medium-sized surfaces, the use of IR cameras has been extended in recent years to various fields of industry and human life, among those that are the inspection and detection of imperfections, defects and cracks in laminated magnetic [8], [9] and ferromagnetic materials [10], in polymers reinforced with carbon fiber [11], in the control of the winding of industrial motors [12], in the supervision of photovoltaic panels [13], in the acquisition and processing of medical images [14], in the detection of cancer incorporating neural networks [15], in the support of medical decisions during neurosurgical interventions [16], in the aerial supervision of renewable energy photovoltaic plants [17], in the detection of faults in electronic systems [18] and in the inspection of resistance spot welding and stamping used in the automotive industry [19]. Other investigations conducted in this regard use 3D thermal images captured dynamically with an IR camera in conjunction with laser triangulation [20].

Lately, different applications involving the use of thermopiles have been made to measure the temperature of people and objects. Some of these applications have been aimed at determining the location of people indoors both to diagnose the perception of human behavior [21] and to manipulate ventilation and temperature control systems [22] using machine learning [23] and images background elimination techniques to avoid non-human thermal sources that interfere with the processing of information [24]. Likewise, recent applications of thermopiles have been directed to people counting [25], the discovery of fungi and spores in the environment [26] and the detection of low-speed vehicles [27] in congestion.

2.MATERIAL AND METHODS

The methodology used to conduct this work was to divide it into two parts: the system hardware and the programming of the computer module.

2.1. System hardware

The system architecture is based on the block diagram shown in Figure 2. It is made up of three elements: the Raspberry Pi 4 computer module, the IR camera, and the HD monitor.

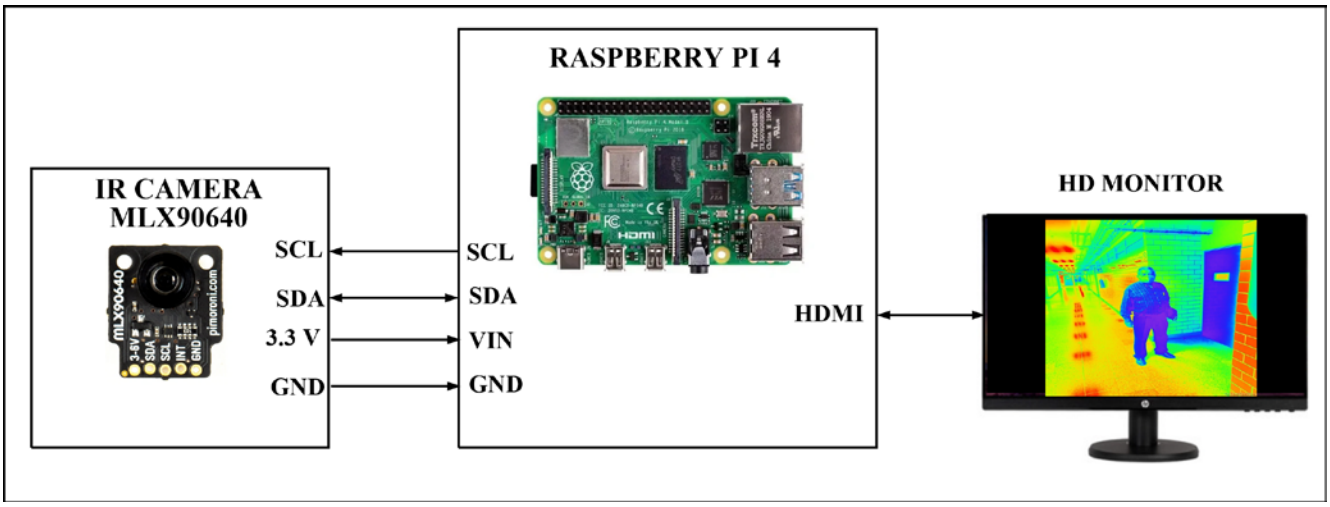

Fig. 2. System architecture.

The Raspberry Pi 4 computer module used in the system has the following hardware resources: Broadcom BCM2711 quad-core 64-bit 1.5 GHz Cortex-A72 processor, GPU, 1 to 8 GB LPDDR4 SDRAM memory, Wi-Fi 802.11 ac, Gigabit Ethernet and Bluetooth 5.0 interfaces, I2C bus, 2 USB 3.0 ports, 2 USB 2.0 ports, 40 General Purpose Input/Output (GPIO) pins, Mobile Industry Processor Interface-Display Serial Interface (MIPI-DSI) monitor port, Camera Serial Interface (MIPI-CSI) camera port, 2 micro HDMI ports for video and sound, and a micro-SD card adapter. This computer module was selected because it is possible to directly connect both the IR camera and the HD monitor to it. Another reason this module was used was that the Raspberry Pi 4 processor is powerful enough to execute the mathematical operations performed by the functions and routines used in programming, which can be done using other embedded modules, but with certain limitations.

The IR camera used was the MLX90640 device. This camera incorporates a 24x32 infrared detector array, providing a resolution of 768 pixels. Each pixel is identified by the row and column in which it is located as indicated in Figure 3. The MLX90640 camera allows capturing images that can have up to 768 different relative temperature values to show the silhouette of people and objects in front of it. You can see in the figure above that the camera pixel map has the origin in the upper right corner. This must be considered when the Raspberry Pi 4 module program reads the temperature value of the sensors, performs their processing and displays the graph on the HD monitor. It is necessary to rotate the position of the pixels so that the origin of the array is in the upper left corner as assumed by most functions used to display graphics and images. The IR camera has an I2C interface whose default working frequency is 400 KHz, or fast mode, and can be configured to transmit information at a maximum frequency of 1 MHz. It provides a field of view (FOV) of 110° and 75° horizontal and vertical, respectively, aimed at short-range measurement applications. The IR camera allows to measure temperature values from -40 ℃ to 300℃, with a resolution of ±1 ℃ and can be configured to use a refresh rate of frames from 0.5 Hz to 64 Hz. This means 64 frames, or samples per second, 64 sps, or 64 Hz. It is compact in size which makes it easy to use in almost any type of application that needs infrared vision.

In this work, the camera was connected to the I2C bus of the Raspberry Pi 4 module, which allows to easily read the information delivered by the camera sensors, process it to convert it to pixel numbers and display the thermal image on

the HD monitor. The I2C address 0x33 that the camera has established by default was used. The HD monitor used was of the type of HP 24y Display LCD 23.8 Led Full HD HDMI.

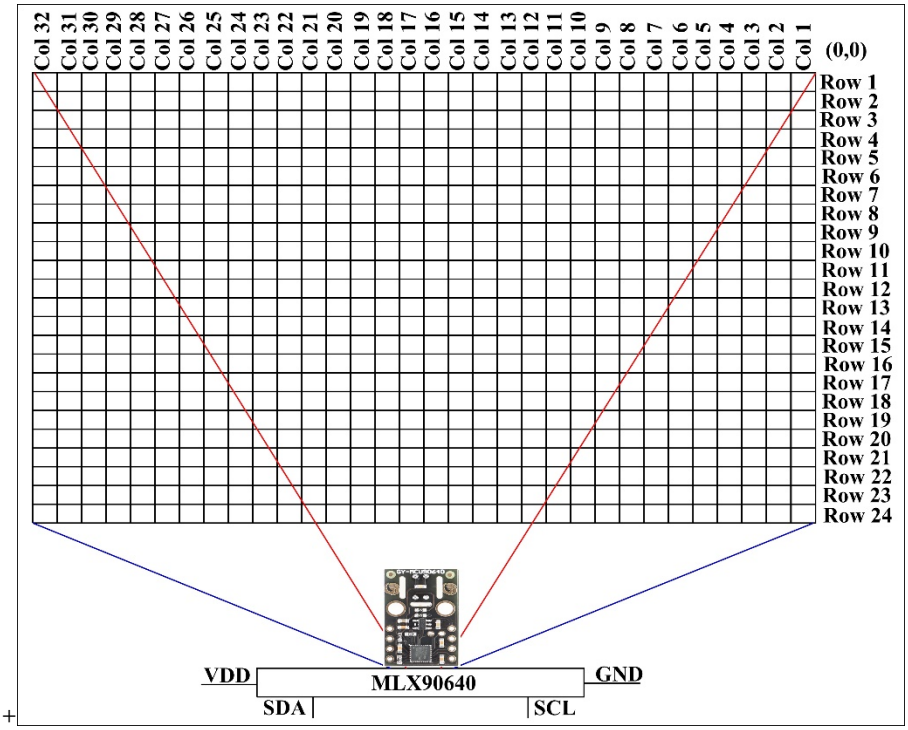

Fig. 3. Distribution of IR sensors of the MLX90640 camera.

2.2. The programming of the computer module

The version 11 of the Raspbian operating system (Raspberry Pi OS) was installed on the Raspberry Pi 4 module. The programming of this module was done in Python using open-source function libraries. To access and control the GPIO terminals used in the communication between the Raspberry Pi and the IR camera, the *RPI*, *GPIO*, *adafruit-blinka* and *adafruit-circuitpython-mlx90640* libraries were used. To access the I2C bus from the Raspbian command line, or from the program, the *i2c-tools* and *smbus2* libraries were used. The smbus2 (System Management Bus) is a protocol layer built on top of the I2C protocol. To invoke the scientific computing functions, the *numpy*, *scipy*, and *matplotlib* libraries were used. The *numpy* (Numerical Python) library is the core of function libraries that make it easy to manage N-dimensional and multidimensional arrays. *Scipy* for Python is built on top of *numpy* and contains mathematical algorithms for manipulating and displaying the arrays defined in *numpy*. *Matplotlib* is a library that allows you to make 2D plots in Python through trends, patterns and correlations. These last three libraries were used in the interpolation conducted before showing the silhouette of people and objects on the HD monitor, as will be explained later. The IR camera was set to communicate with the Raspberry Pi 4 module at a frequency of 400 KHz, which is why the I2C bus clock frequency on the Raspberry was set to use the same frequency value, which was set in the Raspbian boot file */boot/config.txt*.

When you start the main program, it activates and initializes the I2C bus and configures the IR camera to work at 400 KHz and transmit to the Raspberry module two samples per second, or 2 Hz, by calling the functions *busio.I2C*, *adafruit_mlx90640.MLX90640* and *adafruit_mlx90640.RefreshRate.REFRESH_2_HZ*. To display on the HD monitor the graph of the silhouettes with the color of the detected temperature, the *imshow* function contained in the *pyplot* module of the *matplotlib* library was used. This function allows you to display a 2D multichannel RGB image or a single channel grayscale image, based on a data set or a numerical array. Before using *imshow* you must define the size of the plot, or image, in inches, using the *plt.subplots* function. In this way, the next task of the main program is to call the *imshow*

function passing as arguments an array of data with zeros and the range of the color bar to use in the image. The color bar allows you to assign a color to each pixel of the image according to the temperature value read from the corresponding IR sensor. Next, the program initializes the array that stores the temperature value of the 768 pixels, using the *np.zeros* function of the *numpy* library. The IR detector array is 24 x 32, that is, 768 pixels. Subsequently, the program enters a continuous cycle where it reads the temperature values through the I2C bus, using the *mlx.getFrame* function of the *adafruit-circuitpython-mlx90640* library, processes these values and displays the image with the silhouettes.

The processing of the information of the array of temperature values involves some actions before sending the image to the HD monitor. The first one is to perform the rotation of the origin of the array, as indicated above, by means of the *np.fliplr* function. If the image with the data stored in the array up to this point is displayed on the monitor, the image is quite fuzzy and low resolution, so the next task is to perform image interpolation to improve the resolution. Interpolation increases the size of the original image by determining the value of the new data through an algorithm. In this work, the interpolation was done by invoking the *ndimage.interpolation.zoom* function from the *scipy* library. This function uses the S-Spline interpolation method, which determines the color of the new or additional pixels in the array, using all the colors in the image. Invoking this function used a factor of 10 to create an image of 240x320, or 76,800 pixels. Image bounds were then set, and the color bar updated via the *matplotlib* library's *set_clim* and *on_mappable_changed* functions, respectively. The upper limit was set to the color of the pixel with the highest value in the array and the lower limit to the color of the pixel with the lowest value. Once the processing of array values is done, it is time to draw, or display, the image on the HD monitor, which was conducted through the *draw_artist* function. This function is based on the image plotted at the beginning of the program with the *plt.subplots* function.

Even though the program shows the thermal images on the HD monitor, some work conducted in this regard suggests conducting different actions to improve the image processing performance and display a greater number of frames per second. One of them, which was used in this work, because it is implemented in an uncomplicated way and does not require much processing time, is the blitting method. This method consists of showing the complete image the first time it is displayed on the monitor, saving the background of the image and then, in the following frames, updating, or refreshing, the parts of the image that have changed. These actions are conducted in the last part of the program cycle. To save the background, the *fig.canvas.copy_from_bbox* function was used and to refresh the frames, the *fig.canvas.blit* and *fig.canvas.flush_events* functions of the *matplotlib* library were invoked. Figure 4 shows some images shown on the HD monitor obtained by the system.

Fig. 4. Examples of thermal images displayed on the HD monitor.

3.RESULTS AND DISCUSSION

With the system running, three groups of tests were performed. The purpose of the first group was to determine both the performance and the accuracy of the system. The performance was determined by measuring the number of frames per second displayed on the HD monitor. In the execution of these tests, the sampling of two hundred groups of 5 people, on average, was conducted, and the time was obtained by reading the value of the pixels of the camera and after use blitting in the background of the image to displaying it. The test results indicated that 2.4 frames per second are obtained. The

frequency used in I2C communication was 440 KHz. To determine the accuracy of the system, the value of the body temperature of the people of the two hundred groups, obtained in the program by placing the people at 2 meters, was compared against the reference temperature measured by means of a Microlife IFR 100 Hearing Contact Digital Body thermometer. The results of these tests indicated that the accuracy of the system is 99.34% on average.

The second group of tests was conducted to verify the impact of the I2C speed on the number of frames per second obtained. Conducting these tests consisted of sampling the two hundred groups of people configuring the I2C bus with different speeds from 100 KHz, standard speed, to the maximum of 1 MHz. In all the tests 8 sps were configured in the camera. As expected, the higher the I2C speed, the more frames per second are obtained in the image, as indicated in the graph of Figure 5, which is closer to real time. However, it was also found that the higher the speed of the I2C bus, as indicated by the work conducted in this regard, it has the disadvantage that the processor of the Raspberry Pi 4 module heats up faster. No tests were done modifying the number of frames per second delivered by the camera.

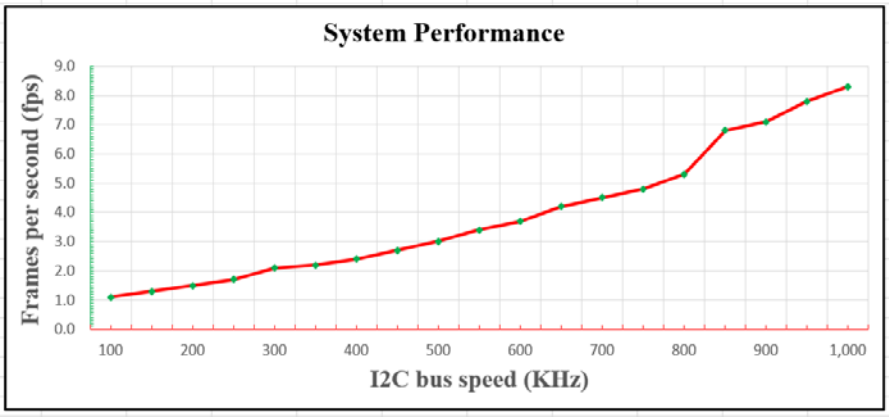

Fig. 5. System performance.

The purpose of the third group of tests was to determine the number of frames per second displayed on the HD monitor without interpolating the array of temperature values read from the camera. In these tests the I2C speed of 440 KHz and 8 sps in the camera was used, 15 frames per second were obtained, however, the image is blurred and of very low resolution. Modifications to the software can be made using other calculation methods and refine system performance. For example, functions from the OpenCV library, specialized in computer vision, can be used to check if it is possible to obtain better results, or use other interpolation methods. This task and the incorporation of a web server in the programming, which allows visualizing the thermal images from the Internet, are future activities for this work.

4.Conclusion

A system was developed that allows thermal images of people and objects captured with an IR camera to be displayed on an HD monitor. The camera used, the MXL90640, is low cost, readily available and used in a variety of applications. It is composed of an array of 24x32 IR sensors and provides the temperature value captured by the 768 sensors. This array of values is interpolated with a factor of 10 to obtain a higher resolution image and to display the temperature of people and objects in the angle of view of the camera on an HD monitor. The reading of the array of sensors and the processing of this information, to obtain the thermal image, was conducted using a Raspberry Pi 4 computer module. The achieved performance of the system was 2.8 frames per second configuring the speed of the I2C bus, used in the communication of the camera with the Raspberry Pi 4 module, with the fast speed of 400 KHz, and configuring in the camera 8 samples per second. These two parameters are values of average magnitude with which images are not obtained in real time, but they are of acceptable quality to detect the temperature in places where there are no large crowds of people.

A factor that contributed to reduce the time and complexity of the development of this work was the use of the free code function libraries *numpy*, *scipy* and *matplotlib* in the information processing. Considering the above and that the cost of the implemented system is more than 20 times lower than the cost of a professional thermal camera, it can be used in various places or facilities, such as laboratories, data centers, access to offices, schools and buildings, among others.

References

[1] M. Canil, J. Pegoraro and M. Rossi, "milliTRACE-IR: Contact Tracing and Temperature Screening via mmWave and Infrared Sensing," in IEEE Journal of Selected Topics in Signal Processing, vol. 16, no. 2, pp. 208-223, Feb. 2022.

[2] L. Estrela and L. Oliveira, "A Prototype of a Termography Equipment," in IEEE Latin America Transactions, vol. 19, no. 12, pp. 2011-2018, Dec. 2021.

[3] S. Karki, H. Ziar, M. Korevaar, T. Bergmans, J. Mes and O. Isabella, "Performance Evaluation of Silicon-Based Irradiance Sensors Versus Thermopile Pyranometer," in IEEE Journal of Photovoltaics, vol. 11, no. 1, pp. 144-149, Jan. 2021.

[4] Z. -H. Wang, G. -J. Horng, T. -H. Hsu, C. -C. Chen and G. -J. Jong, "A Novel Facial Thermal Feature Extraction Method for Non-Contact Healthcare System," in IEEE Access, vol. 8, pp. 86545-86553, 2020.

[5] N. Faulkner, D. Konings, F. Alam, M. Legg and S. Demidenko, "Machine Learning Techniques for Device-Free Localization Using Low-Resolution Thermopiles," in IEEE Internet of Things Journal, Early Access Article, 2022.

[6] R. Yong, G. Zhang, T. Qiang and Y. Jiang, "Highly Sensitive Micromachined Thermopile Infrared Sensor System With Chopper Operational Amplifier," in IEEE Transactions on Electron Devices, vol. 68, no. 9, pp. 4497-4503, Sept. 2021.

[7] Y. Dai, S. Z. Ali, R. Hopper, D. Popa and F. Udrea, "Modeling of CMOS Single Membrane Thermopile Detector Arrays," in IEEE Sensors Journal, vol. 22, no. 2, pp. 1366-1373, 15 Jan.15, 2022.

[8] L. Ferraris, F. Franchini and E. Pošković, "A Novel Thermographic Method and Its Improvement to Evaluate Defects in Laminated and Soft Magnetic Composites Devices," in IEEE Transactions on Industry Applications, vol. 55, no. 6, pp. 5779-5788, Nov.-Dec. 2019.

[9] J. Wu, J. Zhu, Z. Xu and H. Xia, "A DC-Biased Scanning Induction Thermographic System for Characterizing Surface Cracks in Ferromagnetic Components," in IEEE/ASME Transactions on Mechatronics, vol. 26, no. 5, pp. 2782-2790, Oct. 2021.

[10] E. Pošković, L. Ferraris, G. Bramerdorfer and M. Cossale, "A Thermographic Method to Evaluate Different Processes and Assembly Effects on Magnetic Steels," in IEEE Transactions on Industry Applications, vol. 58, no. 3, pp. 3405-3413, May-June 2022.

[11] Z. Luo et al., "Enhanced CFRP Defect Detection from Highly Undersampled Thermographic Data via Low-Rank Tensor Completion-Based Thermography," in IEEE Transactions on Industrial Informatics, Early Access Article, 2022.

[12]Y. Peng et al., "Joint Scanning Electromagnetic Thermography for Industrial Motor Winding Defect Inspection and Quantitative Evaluation," in IEEE Transactions on Industrial Informatics, vol. 17, no. 10, pp. 6832-6841, Oct. 2021.

[13] X. Guo and J. Cai, "Optical Stepped Thermography of Defects in Photovoltaic Panels," in IEEE Sensors Journal, vol. 21, no. 1, pp. 490-497, 1 Jan.1, 2021.

[14] J. A. Leñero-Bardallo, R. De la Rosa-Vidal, R. Padial-Allué, J. Ceballos-Cáceres, Á. Rodríguez-Vázquez and J. Bernabéu-Wittel, "A Customizable Thermographic Imaging System for Medical Image Acquisition and Processing," in IEEE Sensors Journal, Early Access Article, 2022.

[15] M. A. S. A. Husaini, M. H. Habaebi, S. A. Hameed, M. R. Islam and T. S. Gunawan, "A Systematic Review of Breast Cancer Detection Using Thermography and Neural Networks," in IEEE Access, vol. 8, pp. 208922-208937, 2020.

[16] J. Müller et al., "Registration and Fusion of Thermographic and Visual-Light Images in Neurosurgery," in IEEE Transactions on Biomedical Circuits and Systems, vol. 12, no. 6, pp. 1313-1321, Dec. 2018.

[17] S. Gallardo-Saavedra, L. Hernández-Callejo and O. Duque-Perez, "Image Resolution Influence in Aerial Thermographic Inspections of Photovoltaic Plants," in IEEE Transactions on Industrial Informatics, vol. 14, no. 12, pp. 5678-5686, Dec. 2018.

[18] G. R. Hernández, M. A. Navarro, N. Ortega-Sánchez, D. Oliva and M. Pérez-Cisneros, "Failure Detection on Electronic Systems Using Thermal Images and Metaheuristic Algorithms," in IEEE Latin America Transactions, vol. 18, no. 08, pp. 1371-1380, August 2020.
[19] S. Guo, D. Wang, Z. Feng and W. Guo, "UIR-Net: Object Detection in Infrared Imaging of Thermomechanical Processes in Automotive Manufacturing," in IEEE Transactions on Automation Science and Engineering, Early Access Article, 2022.
[20] B. Deng et al., "Active 3-D Thermography Based on Feature-Free Registration of Thermogram Sequence and 3-D Shape Via a Single Thermal Camera," in IEEE Transactions on Industrial Electronics, vol. 69, no. 11, pp. 11774-11784, Nov. 2022.
[21] N. Gu, B. Yang and T. Zhang, "Dynamic Fuzzy Background Removal for Indoor Human Target Perception Based on Thermopile Array Sensor," in IEEE Sensors Journal, vol. 20, no. 1, pp. 67-76, 1 Jan.1, 2020.
[22] L. Wu and Y. Wang, "Compressive Sensing Based Indoor Occupancy Positioning Using a Single Thermopile Point Detector With a Coded Binary Mask," in IEEE Sensors Letters, vol. 3, no. 12, pp. 1-4, Dec. 2019.
[23] N. Faulkner, F. Alam, M. Legg and S. Demidenko, "Device-Free Localization Using Privacy-Preserving Infrared Signatures Acquired From Thermopiles and Machine Learning," in IEEE Access, vol. 9, pp. 81786-81797, 2021.
[24] T. Zhang, B. Yang and N. Gu, "Application of A Priori Map in Dynamic Background Removal for Indoor Human Perception Using a Thermopile Array," in IEEE Sensors Journal, vol. 22, no. 2, pp. 1154-1162, 15 Jan.15, 2022.
[25] E. Hagenaars, A. Pandharipande, A. Murthy and G. Leus, "Single-Pixel Thermopile Infrared Sensing for People Counting," in IEEE Sensors Journal, vol. 21, no. 4, pp. 4866-4873, 15 Feb.15, 2021.
[26] S. Pham, A. Dinh and K. Wahid, "A Nondispersive Thermopile Device With an Innovative Method to Detect Fusarium Spores," in IEEE Sensors Journal, vol. 19, no. 19, pp. 8657-8667, 1 Oct.1, 2019.
[27] Y. Zhang and B. Yang, "Traffic Flow Detection Using Thermopile Array Sensor," in IEEE Sensors Journal, vol. 20, no. 10, pp. 5155-5164, 15 May15, 2020.