"Seeing" Temperature - Integration of IR camera into the HoloLens 2

Eric Enzler, Gianluca Moro, Markus Pobitzer, Claudio Cannizzaro, Benjamin Suter*
Eidgenössische Technische Hochschule Zürich

Abstract

"Seeing" temperature can offer enormous advantages suitable for a diverse variety of applications, one of them being efficient contactless screening for COVID-19 symptoms in crowded areas. Instead of successively measuring the temperature of every individual, AR applications in combination with thermal cameras could highlight and track potential risk bearers showing COVID-19 symptoms such as abnormally high body temperature. We provide a system that modifies the physical build of a Microsoft HoloLens 2 such that it holds Hardware components capable of measuring temperature. Furthermore, it uses a TCP connection to send data from a Raspberry Pi to the HoloLens, processes this data, and provides different modes for the user to visualize the thermal data.

1. Introduction

Ever since the beginning of 2020, the earth and its way of life have been dictated by the COVID-19 outbreak. People have started to worry more about their health and have taken action against the mass-spreading of viruses. Various technologies and machines have been developed to screen body temperatures of individuals to detect risk bearers showing COVID-19 symptoms. In some countries, such machines are used to permit or allow entry to certain office buildings. They most often work by solely measuring temperature. In this work, we take existing machines to measure and display body temperature and combine them with the advancements of mixed reality technology to display the user the thermal information directly into the AR visor.

Thermal imaging in combination with augmented reality (AR) is not only interesting for the medical field. The advantage of thermal imaging with infrared (IR) sensors is that it can neglect the visual interference that a human would have. An example of this is night vision or vision through a fog. If we combine these features with AR technology, we can extend the vision capability of a person drastically and display thermal information directly to the visual feed. The US army is developing a new generation of Night Vision goggles (NVGs) that use thermal imaging coupled with AR features [15]. A startup called Longan Vision is creating an AR Visor combined with thermal cameras for firefighters, to give them the ability to see through smoke in a burning building [17].

*Supervisor, Nanoparticle Systems Engineering Laboratory, ETHZ

2. Related Work

Our work is based on perceiving temperature in the world with a special interest in body temperature, cameras that can capture the thermal radiation of the objects corresponding to the temperatures, and the visualization of said temperature in mixed reality (MR). Cameras that can create thermal images based on the energy waves of the thermal radiation are called forward-looking infrared (FLIR) cameras or thermal cameras. To visualize the thermal data in MR, a head-mounted display (HMD) is used, other options include smartphones and tablets.

Body temperature. The normal human body temperature is around 37 °C. Different body parts can have higher or lower temperatures because of a transfer of heat to and from the core. When a person has a fever the core temperature rises to 38-40 °C, temperatures above 41 °C can lead to critical conditions. [1]. Hewlett et al. showed that infrared thermal detection systems have the potential to screen persons for febrile illness [7].

Temperature visualization. In this section we consider related work incorporating visualization of temperatures using an HMD.

A closely related work is from Erickson et al. where the authors investigate the connection between felt temperature and visual representation of temperature through color or holograms [5]. They implemented a thermal vision view similar to ours but using two thermal cameras. The HoloLens is used as the HMD but the whole system is not used as standalone since everything gets streamed to the HoloLens via Unity holographic remoting. Through this, they bypass the hardware limitations of the HMD.

Schönauer et al. presented a system that can help firefighters map out a building and find stuck people based on their body temperature in extreme situations such as fire and smoke [16]. They too used quite extensive hardware to support the processing of the system. In addition, the used thermal camera has a high refresh rate of 50 Hz that enables them to create a thermal live preview. For thermal cameras that use a lower refresh rate such as a live stream over the user's view is not possible without breaking the feel of immersion or worse without making the user dizzy.

Knierim et al. showed an example of visualizing thermal flux in MR [11, 12]. They used a stationary rod to visualize the temperature and the infrared camera was also fixed. Since it is a known and stationary object it makes the capturing of the thermal data easier. The worn HMD knows the position in space of the rod through a marker under the object. The work is a good example of how thermal data can be incorporated into MR.

3. System Overview

3.1. Hardware

AR device. For our project, we used a Microsoft HoloLens 2 [13]. It is a mixed reality headset that has a high resolution coupled with head and eye-tracking. The 6 DoF tracking allows the headset to track its position in the real world. With the built-in depth sensor, it can create a world mesh around the space of the user. The built-in computation device and especially their custom build holographic processing unit is powerful. However, we always have to consider that this is a mobile device and the provided resources are limited.

Thermal Camera. The camera is a LEPTON®3.5 from FLIR[1], a small camera that captures thermal images of 160x120 pixels size. It has a temperature accuracy of $\pm$ 5°C. The camera weighs 0.9 grams and is sitting on a FLIR Lepton® Camera Breakout Board v2.0 to easily connect it to the Raspberry Pi and power the camera via a cable. When we talk about the physical build of our application, we assume that the camera and its board are one piece.

Minicomputer. The infrared camera is connected to a Raspberry Pi 4 via a cable. The Raspberry Pi is a cheap, credit-card-sized computer. In our work, it is used to process the captured infrared data, convert it to Celsius degrees and then send the temperature data to the HoloLens.
To make the Raspberry Pi portable, we added a PiJuice battery to it. Otherwise, the minicomputer has to be constantly powered by cable.

3.2. Physical Build

Camera. To make the build portable, we had to come up with a system that would allow us to stick the IR camera and the Raspberry Pi directly to the HoloLens. In a first step, we 3D printed a mount for the camera, attached the camera to it via four small screws, and glued everything on top of the front enclosure of the HoloLens as in Fig. 1b. Since the IR camera and its mount weigh little to nothing, the user does not experience any balancing issues. Note that in this state, the camera is rotated by 90 degrees.

Raspberry Pi. The Raspberry Pi turned out to be more difficult. First, we had to create a casing for it. Commercially available cases were too small because we have additional height from the PiJuice battery. For this reason, we took an existing 3D model of a case [3], extended its height, and added more air circulation openings. We printed it at ETH and it rested a little bit more than six hours in the printer. Over the span of the course, we redesigned the case once to optimize it for our input and output cables as seen in Fig. 1a.
We tried out different places to attach the Raspberry Pi to the HoloLens and decided to mount it on top of the rear cover because there, the influence of the additional weight was minimized and the user was still able to operate the HoloLens to its full potential. Furthermore, the contact surface at this part is nearly flat. This gives a bit more stability to our attached case.

3.3. IR Camera Module Implementation

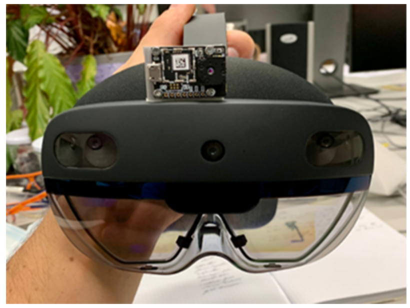
IR Data. To convert the data from the infrared camera to Celsius degrees, we are using a Python script running on the Raspberry Pi. We have modified an implementation from the "purethermal1-uvc-capture" [10] GitHub repository. The author permitted us to use the code for our application. The basic idea of the conversion is to capture the infrared energy emitted from the environment and for each pixel, the amount of photons is scaled to a 14-bit integer which represents the temperature in Kelvin. Finally, we just need to convert the temperature from Kelvin to Celsius.
The resulting data is a 2D array of size 160×120 containing the temperature of each pixel in Celsius degrees.

TCP Data Transfer. We are using a Python script that runs a TCP server on the Raspberry Pi to which the HoloLens application connects. The app is listening on the connection for the temperature data. The TCP server gets a single frame from the IR camera, as described above, and sends the data in JSON format to the HoloLens. The HoloLens app can then deserialize the JSON to get the actual temperature data.

Image Alignment. Since we are accessing two camera feeds, we have two pictures of different sizes and angles. To tackle this problem, we decided to use feature matching to create a homography matrix. We use this matrix to warp an image from the HoloLens image space to the IR camera

[1]FLIR systems, US company

(a) 3D printed Raspberry Pi case



(b) IR camera attached to the HoloLens

(c) The final physical build

Figure 1. Physical Build

image space. This by itself does not sound too complicated, but since the two camera spaces have different image sizes and modes, we need to consider some points.
First of all, because the images have very different resolutions, namely the HoloLens image has a resolution of 3904×2196 pixels and the thermal image only 160×120, we resize the images. We downscale the HoloLens image to 1920×1080 and upscale the thermal image to 492×369. We also convert both images from RGB to grayscale. Furthermore, because the IR camera is mounted rotated by 90° to the HoloLens, we also rotate the thermal image by 90° counterclockwise so that both images are oriented the same way.
After that, our approach uses SIFT[2] features to find corresponding feature points in both images. We compared the algorithm with ORB[3] features but did get fewer and more faulty matches with ORB. Finally, we use the RANSAC[4] algorithm to select the best matches and create a homography matrix by solving the equation:

$$H = QP^T(PP^T)^{-1} \tag{1}$$

where P is the matrix of selected points in the HoloLens image and Q is the matrix of the same points in the IR camera image. We can then apply a homography transformation to any pixel in the HoloLens image using H and get the resulting pixel in IR camera coordinates. In the current build, we store H hardcoded on the HoloLens application so that the TCP traffic is minimized while using the app.

3.4. HoloLens Implementation

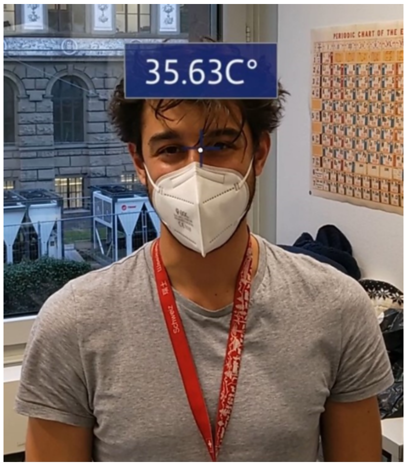
As the main development tool, we used Unity and C# [18]. We developed our project under Unity version 2020.3.20f1, for the setup we used the Mixed reality feature Tool provided by Microsoft that made the Unity setup easier. This also included the installation of the Mixed reality toolkit (MRTK) in Unity that provides us with the functionalities needed to interact with the HoloLens 2 [14]. Our project includes the JSON .NET For Unity package [8]. This package is needed for data transfer purposes.

HoloLens App. The AR applications were designed to fulfill the screening of temperature as the main application. Therefore, the App directly launches into the ”Crosshair” mode that mimics a thermometer. The second mode called ”Imaging” mode lets the user display the thermal image in the world. We also thought about adding a third mode that can automatically detect faces but this is not included in the current version. The reason for not including the face detection mode will be explained later.
We will now discuss the implementation of the different modes in a bit more detail.

Crosshair mode. When the user launches the app or selects this mode, a reticle gets displayed in the center of the view. This is the area where we get the temperature from. A panel with the temperature data is displayed above the crosshair. Both holograms follow the user's head movements. The result can be seen in Figure 2a. To implement this we update the position and rotation of the game objects according to the camera position that corresponds to the head of the user and put the holograms further down the z-Axis. To extract the temperature value at the right position we use the homography matrix from equation 1 that maps pixel coordinates from the RGB camera of the HoloLens to the corresponding value of the IR camera. The right pixel position in the RGB image was found out manually, note that this does not consider the depth of the object currently measured and could lead to small inaccuracies. This is the main mode that could replace the thermometer currently used when doing entrance checks in hospitals, airports, and events.

Imaging mode. Switching to the imaging mode deactivates the crosshair mode and lets the user create a thermal

[2]Scale-invariant feature transform
[3]Based on the FAST feature detector and BRIEF binary descriptor
[4]Random sample consensus



(a) The Crosshair mode used on a person wearing a mask.

(b) Two people sitting on a chair without any holograms.

(c) The same two people as seen in Figure 2b but now with the thermal overlay from the Imaging mode.

Figure 2. Crosshair and Capture mode as seen by the user wearing the HoloLens.

image. The thermal image gets drawn on a plane and has the right position and rotation in the world. A result of this can be seen in Figure 2c. We decided to go with picture-only mode instead of a video sequence since the IR camera provides only 7 frames per second (fps) whereas a hologram should normally run at 60 fps.

Thermal image. From the IR camera, we get an image frame with temperature data. We visualize the thermal data on a plane and use a custom build shader for the right pixel position and color. To position the plane in the real world we use the camera to real-world matrix provided by the PhotoCapture API [9]. Using this matrix, we can again map pixels of the RGB image to the IR camera data. When mapping temperature data to color, we chose to give the lowest temperature the color blue with RGBA (0, 0, 1, 1), the highest the color red with RGBA (1, 0, 0, 1) and interpolate the other colors in between to create a false-color image. For this, we use Unity's Color. Lerp function. In the end, we tell the shader at which pixel to set the specific color. Note that the holograms use an additive color model, this means that black will appear as transparent but this is not that important for us. Our shader is based on a Unity post [6]. An example with min and max temperature can be seen in Figure 3.

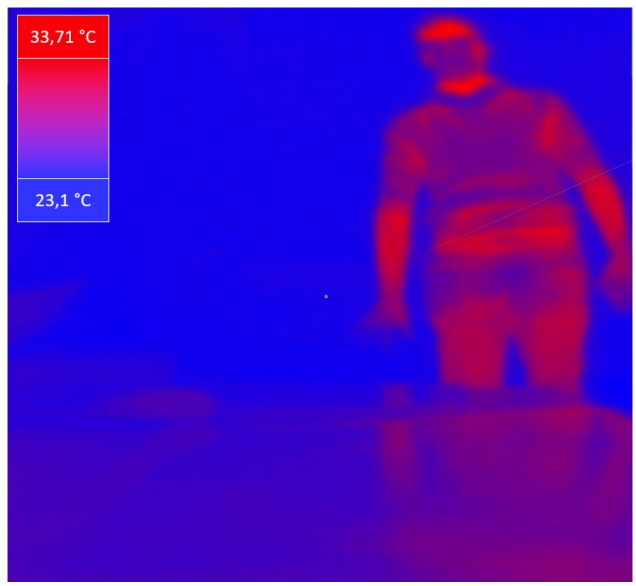


Figure 3. Thermal image of a person with color legend

4. Experiments

4.1. Portability

We have walked around outside and inside with our system to see if the physical build was attached properly to the HoloLens and if it would have any disadvantages for the user. The only noticeable drawback is that the user needs to be more careful when adjusting the HoloLens to the headsize with the wheel at the back because users with big fingers cannot reach between the wheel and the Raspberry casing.

4.2. Temperature Accuracy

In order to measure the accuracy of our system, we measured different objects with the crosshair mode of our Unity application and compared the results with the measurement of a more advanced and accurate thermal camera[5]. The advanced camera has an accuracy of $\pm$ 2°C at room temperature. The results stated that our systems would on average

[5]Fluke Ti110

return a temperature that was 2°C lower than the one of the advanced thermal camera. We have therefore adapted our system by adding 2°C to the output temperature. This is especially useful for our application since we expect fever to show an abnormally high body temperature. This allows an inexperienced user to see more familiar results when scanning persons with abnormally high body temperatures.

5. Limitations

In this section, we list some of the limitations of our work and possible future improvements. Furthermore, we state the legal limitations of our approach - and therefore neglect the ethical limitations.

5.1. Hardware

Unfortunately, our approach is limited heavily by the hardware that we are using. We have listed the most cumbersome problems in this section.

HoloLens 2 The used HMD is one of the limiting factors. Despite being powerful for such a device, the stability and performance of the application were not always there. Often the fps dropped below 60 and this should not happen for holograms. Optimizing for performance is quite hard since the debugging information was rather obscure.
When using the camera of the HoloLens an fps drop was noticeable although everything happens asynchronously as mentioned in the documentation.

Battery Life. Even though the Raspberry Pi has a PiJuice battery connected to it, which should have a battery life of around four hours, we experienced that we ran out of battery after only about 30 minutes. We suspect that the reason for this is the infrared camera module which also runs off the Raspberry Pi's battery. Another reason is the TCP connection script, which runs constantly while running the application. A positive thing is that we do the bulk of the processing on the HoloLens. This saves a bit of power.
To extend the battery life, we could connect a power bank to the Raspberry Pi or buy a larger battery. The first approach would require us to attach even more things to the HoloLens, which is essentially already out of space. The second one is a redesign of our Raspberry casing. Both ideas will add more weight to the HoloLens.

Direct connection. Since the USB port on the HoloLens is only for charging the device, we are not able to plug in the camera directly into the HoloLens and access its data stream. The Bluetooth function of the AR device does not help either, since the IR camera does not support Bluetooth features. This implies that we can neither get rid of the Raspberry Pi, nor plug the Raspberry Pi into the HoloLens via the USB port. However, it is possible to connect the two devices via Bluetooth by using a specific API[6]. We decided to use a TCP connection for the sake of simplicity.

5.2. Software

TCP Connection. Since we decided to use a TCP connection for the data transfer, we always need to be connected to the internet. Furthermore, we need to know the IP address of the Raspberry Pi at any time. Because the IP address is hardcoded in the script, we need to modify the connection script when changing the network. We cannot find the correct IP address from the HoloLens directly as this would require having some sort of public known address like a server. By using a server, we could modify the script such that the TCP connection could be instantiated as soon as both the HoloLens and the Raspberry Pi are connected to a network. Furthermore, it would allow to push and pull the thermal data from the IR camera directly to the server. A drawback to this method would be the latency, which is already fairly high because of the IR camera. Furthermore, this requires even more hardware components. Finally, the developer would have to look at the legal restrictions of saving medical information on a server.

Mapping. The provided mapping from the HoloLens frame to the IR camera frame is hardcoded at this point. We planned to provide a function that would allow the user to renew the mapping when starting the AR application. The mapping might become worse over time because the IR camera could possibly shift if the screws loosen up or the user plays around with the mount. Our team decided to not implement such a feature because there need to be a lot of things considered when doing such a mapping. Since we are comparing tiny thermal pictures with high-resolution pictures, we need to consider a few things to get a useable mapping. For example, we need a black background that is not too hot when taking the baseline image. If the background is not black, we will not get enough good features to create a faultless mapping.

Face detection. We have come up with the idea to apply face detection on the thermal image. In Python, we could use open source code and apply the detection algorithm directly to the thermal image. This approach did not work very well, since the thermal image has not enough features to properly detect faces. A possible solution to this would be to use a machine learning pipeline to train a deep network on thermal faces.
A different approach to this problem could be to send the HoloLens image to the Raspberry Pi, search for faces and return the found rectangles - or simply send the thermal

[6]Bluetooth RFCOMM API

data inside the found face rectangles. The drawbacks of this method are the additional network traffic, its latency, and the power drain of the PiJuice battery.

The next approach was tracking faces with the HoloLens and then applying the Homography matrix 1. This would allow us to determine the exact surface temperature of a face and subsequently just display the temperature above a face. There exist a sample of this by Microsoft which tracks a face with the HoloLens2 [2], but this does not work as is within a Unity application. This is because Unity uses its own C# compiler which does not support the newer .NET standards used by the sample. The next idea we had, was to implement this ourselves using OpenCV. This also did not work, because there are no open-source versions of OpenCV available for Unity. [18]

An additional concern that came up after testing with the Microsoft sample was that simple face tracking did not work flawlessly. Since people are required to wear masks almost everywhere indoors, the sample had serious problems when trying to detect masked faces. A solution to this could be to modify existing face detection algorithms or use machine learning to create a deep network that detects faces wearing masks.

5.3. Legal Limitation

In Switzerland, gathering medical information such as body temperature is restricted by law [4]. Our application is for example not allowed to save the collected temperature data of a person. This is no problem because we only display the current temperature and do not save anything in our system. Furthermore, we are not allowed to scan a person's body temperature against their will. Finally, we are not allowed to make any assumptions about the health status of a person we scanned. This is restricted to medical personnel.

6. Conclusion

This paper presents the combination of hardware and software elements being added to an AR device to make temperature information visible to a user. We think that our proposed system has the potential to monitor persons to detect febrile illness. However, for accurate medical results, an approved thermal camera should be used. The portable and light design makes it easy to wear for prolonged periods. Since we did not want to be bound by some additional hardware, limitations are still there. The HoloLens 2 was not specifically designed to work together with specialized hardware. The limitations could be overcome with new generations of the HoloLens that may include more computing power, AR-specific optimizations, and better hardware integration support.

References

[1] John Bligh, Karlheinz Voigt, HA Braun, K Brück, and G Heldmaier. *Thermoreception and temperature regulation*. Springer, 1990. 1

[2] Raymond Chen. Basic face tracking sample, 2021. URL: https://docs.microsoft.com/de-de/samples/microsoft/windows-universal-samples/basicfacetracking/. 6

[3] 219 Design. Case for raspberry pi 4, 2020. URL: https://grabcad.com/library/case-for-raspberry-pi-4-1/details?folder_id=8022281. 2

[4] EDÖB. Datenschutzrechtlicher rahmen bei der eindämmung des coronavirus, 2020. URL: https://www.edoeb.admin.ch/edoeb/de/home/aktuell/aktuell_news.html#1848566086. 6

[5] Austin Erickson, Kangsoo Kim, Ryan Schubert, Gerd Bruder, and Greg Welch. Is it cold in here or is it just me? analysis of augmented reality temperature visualization for computer-mediated thermoception. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 202–211, 2019. 1

[6] Brandon Fogerty. Holographic photo blending with photocapture. 4

[7] Angela L. Hewlett, Andre C. Kalil, Rahman A. Strum, Wesley G. Zeger, and Philip W. Smith. Evaluation of an infrared thermal detection system for fever recognition during the h1n1 influenza pandemic. *Infection Control & Hospital Epidemiology*, 32(5):504–506, 2011. 1

[8] Dustin Horne. 3

[9] Microsoft Keveleigh. Photo video camera in unity - mixed reality. 4

[10] Kurt Kiefer. Purethermal uvc capture examples. URL: https://github.com/groupgets/purethermal1-uvc-capture. 2

[11] Pascal Knierim, Francisco Kiss, Maximilian Rauh, and Albrecht Schmidt. Tangibility is overrated: Comparing learning experiences of physical setups and their virtual equivalent in augmented reality. In *19th International Conference on Mobile and Ubiquitous Multimedia*, MUM 2020, page 299–305, New York, NY, USA, 2020. Association for Computing Machinery. 2

[12] Pascal Knierim, Albrecht Schmidt, and Thomas Kosch. Demonstrating thermal flux: Using mixed reality to extend human sight by thermal vision. In *19th International Conference on Mobile and Ubiquitous Multimedia*, MUM 2020, page 348–350, New York, NY, USA, 2020. Association for Computing Machinery. 2

[13] Microsoft. Hololens 2 – overview, features and specs: Microsoft hololens. URL: https://www.microsoft.com/en-gb/hololens/hardware. 2

[14] Microsoft Polar-Kev. Mrtk. 3

[15] The Washington Post. The army's latest night-vision tech looks like something out of a video game, 2021. URL: https://www.washingtonpost.com/technology/2021/05/24/army-night-vision-goggles/. 1

[16] Christian Schönauer, Emanuel Vonach, Georg Gerstweiler, and Hannes Kaufmann. 3d building reconstruction and thermal mapping in fire brigade operations. In *2013 IEEE Virtual Reality (VR)*, pages 1–2, 2013. 2

[17] IEEE Spectrum. Startup's thermal imaging and ar system for firefighters joins the covid-19 fight, 2020. URL: https://spectrum.ieee.org/startups-thermal-imaging-and-ar-system-for-firefighters-joins-the-covid19-fight. 1

[18] Unity Technologies. 3, 6