Real-time Thermal Medium-based Breathing Analysis with Python

Breawn Schoun
Computer Science
Comcast Media & Technology Center
University of Colorado Denver
breawn.schoun@ucdenver.edu

Shane Transue
Computer Science
Comcast Media & Technology Center
University of Colorado Denver
shane.transue@ucdenver.edu

Min-Hyung Choi
Computer Science
Comcast Media & Technology Center
University of Colorado Denver
min.choi@ucdenver.edu

ABSTRACT

Respiration monitoring is an important physiological measurement taken to determine the health of an individual. In clinical sleep studies, respiration activity is monitored to detect sleep disorders such as sleep apnea and respiratory conditions such as Chronic Obstructive Pulmonary Disease (COPD). Existing methods of respiration monitoring either place sensors on the patient's body, causing discomfort to the patient, or monitor respiration remotely with lower accuracy. We present a method of respiratory analysis that is non-contact, but also measures the exhaled air of a human subject directly through a medium-based exhale visualization technique. In this method, we place a thin medium perpendicular to the exhaled airflow of an individual, and use a thermal camera to record the heat signature from the exhaled breath on the opposite side of the material. Respiratory behaviors are extracted from the thermal data in real time using Python. Our prototype is an embedded, low-power device that performs image and signal processing in real-time with Python, making use of powerful existing Python modules for scientific computing and visualization. Our proposed respiration monitoring technique accurately reports breathing rate, and may provide other metrics not obtainable through other non-contact methods. This method can be useful for medical applications where long-term respiratory analysis is necessary, and for applications that require additional information about breathing behavior.

KEYWORDS

Respiration, exhale, thermal, medium, Python, medical

ACM Reference Format:
Breawn Schoun, Shane Transue, and Min-Hyung Choi. 2017. Real-time Thermal Medium-based Breathing Analysis with Python. In *PyHPC'17: PyHPC'17: 7th Workshop on Python for High-Performance and Scientific Computing, November 12–17, 2017, Denver, CO, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3149869.3149874

1 INTRODUCTION

Respiration rate is an important indicator of an individual's health that can be comfortably monitored over time. Abnormal respiration rates can indicate serious medical problems in a patient. In overnight studies, this vital sign is monitored to detect and diagnose sleep disorders such as sleep apnea, a potentially fatal medical condition where the patient may stop breathing for sort periods of time. Several methods of long-term respiration monitoring are used in clinical sleep studies [5], and are generally categorized as either contact or non-contact. Contact respiration monitoring involves placing sensors on the patient's body to measure breathing rate. Common methods include placing ECG electrodes on the patient's body [9], putting thermistors in the patient's nose [12], and having the patient wear an abdominal strain-gauge transducer [10]. These methods measure breathing rate directly and have a high rate of accuracy. However, contact methods are uncomfortable, and alter the natural breathing of the patient. For this reason, several non-contact methods of respiration monitoring have been proposed. These methods strive to preserve natural breathing by measuring respiration remotely, using sensors such as cameras [13] [1] [7] [17], volumetric sensing [14], microphones [11], and radar [8]. These methods improve patient comfort, but are often subject to environmental conditions, extraneous movement, or variability in the physical traits of the patient.

In this paper, we propose a novel non-contact respiration monitoring system that monitors a patient's breathing patterns in real-time. To monitor respiratory behavior, we place a thin medium perpendicular to the exhaled airflow of the individual and record the thermal signature on the medium with a thermal camera, and then use Python to process the thermal data and display the extracted information. Our current prototype provides a graphical user interface that processes thermal images from a low-cost thermal camera and displays the respiration rate in breaths per minute (BPM). This software has been tested on a Raspberry Pi (3) to show that our solution is capable of processing this data in real-time on low-power compute-bound hardware. We aim to develop a medical device for real-time respiration monitoring using a Raspberry Pi and an affordable thermal camera for in-clinic use.

Our proposed method of respiration monitoring combines the strengths of contact and non-contact techniques by measuring respiration both directly and remotely. This results in higher measurement accuracies, but without the cost of causing discomfort or distress to the patient. This method also has the potential to provide other metrics not obtainable through other respiration monitoring methods, such as breathing strength, nose to mouth distribution, and tidal volume estimates. Nose to mouth distribution behavior is a metric difficult to obtain from non-contact methods, and because nasal obstruction is known to increase the risk of sleep-disordered breathing problems, this measurement is highly valuable in a wide array of respiratory and sleep related studies.

PyHPC'17, November 12–17, 2017, Denver, CO, USA

https://doi.org/10.1145/3149869.3149874

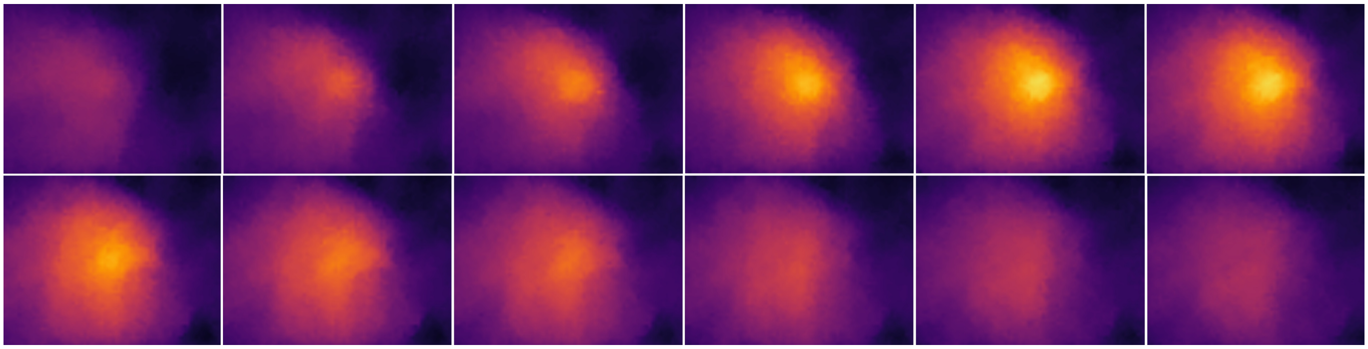


Figure 1: The image sequence above illustrates the recorded thermal projection of the patient's breath during one complete exhale. The images 1 - 6 (top-row) illustrate the increasing heat distribution due to the exhale projected onto the surface of the medium and images 7-12 (bottom row) illustrate the dissipation of the exhale heat. This result is based on the thermal material properties of the selected medium, the exhale strength, and thermal signature transfered from the exhale to the medium.

2 RELATED WORK

Numerous contact and several non-contact respiration monitoring methods have been proposed. Common non-contact methods that use normal cameras to extract the up-and-down movement of the patient's chest [13] have been widely studied. This method is simple to implement and low-cost, but the respiration signal is degraded whenever the patient moves and is highly subject to posture. A similar method that aims to address these problems uses facial tracking to infer the location of the patient's abdomen, making it more robust against patient movement [1].

Other non-contact methods take inspiration from the thermistor-in-nose technique, and uses a thermal camera to measure the change in temperature of the patient's skin underneath their nose [7] [17]. This method disregards exhaled air from the mouth, making it unsuitable for sleep apnea studies since sufferers usually breathe through their mouth. 3D depth imaging sensors have also been used to monitor respiration by measuring the volumetric deformation of a patient's chest [14]. This method has the added benefit of being able to infer breathing volume from chest displacement and visualize complex chest deformations. However, this method can suffer inaccuracies due patient movement and clothing that interferes with the reconstruction. Another similar method uses doppler radar to measure chest movement [8]. However, unlike 3D scanning methods, this method can only measure one point on the body of the subject at a time. Due to variance in physical characteristics and how people breathe, successful measurements depend on determining the correct correlation between chest movement and respiratory behaviors.

A method similar to our proposed technique has been used to study nasal exhale activity [3] using non-contact visualization. In this technique, the patient exhales through their nose onto a thermochromatic liquid crystal film, and signature images are taken on the other side using a film camera. Our method improves upon this by using a thermal camera and image processing, neither of which were widely available during these initial experiments. The proposed technique has also been used to visualize gases for other applications, such as visualizing airflow from air diffusers [4] and examining the temperature distribution of air coming out of cooling passages [16], both of which are analogous to our domain.

3 METHOD

Visualizing exhale patterns due to the thermal intensity of the breath with respect to the cooler surrounding environment is both an intuitive and straight-forward process for understanding breathing behaviors such as rate, strength, and the nose/mouth distribution of a natural exhale. Our method is based on this premise and implemented through the process of projecting an individual exhale onto a projection *medium* that is used to visualize the thermal distribution of the exhale pattern. The intuitive result of this process is shown in Figure 1. This allows us to use thermal imaging to record exhale events during the normal breathing process as they are projected onto a surface medium and then analyze the respiratory behaviors based on the resulting thermal distribution, as outlined in Figure 2.

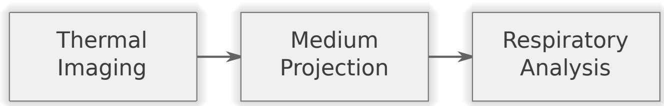


Figure 2: Method overview for medium projection-based thermal respiratory analysis. This method provides a practical link between affordable thermal imaging devices, widely available project mediums, and effective respiratory monitoring for children and long-term studies.

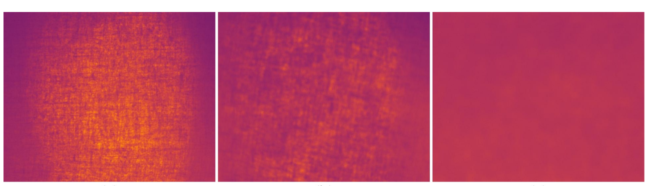
The implementation of our method is based on several core advantages provided by Python including: complete embedded device control (Raspberry Pi) written exclusively in Python, rapid prototyping for experimental design, and real-time image and signal processing within compute-bound hardware. These aspects vital for rapid experimental and design changes that are required for integrating different algorithms into our prototype. We use the Python OpenCV (**Open** Source **C**omputer **V**ision Library) [2] and NumPy (**Num**erical **Py**thon) [15] modules for image processing and statistical analysis, the SciPy (**Sci**entific **Py**thon) module [6] for signal processing, and PyQtGraph and PyQt5 for visualization. These Python libraries provide the foundation of both the visualizations and the signal processing required to extract respiratory behaviors using an inexpensive, compute-bound, mobile device of our implemented prototype.

3.1 Thermal Imaging

Detecting the physical exhale process through the use of thermal imaging provides an intuitive method for visualizing an individual's breathing characteristics. This basic principle is based on the observation that the general indoor ambient temperature is commonly much lower than an individuals general exhale temperature, thus thermal cameras can be used to detect slight changes in temperature related to breathing behaviors. However, the exhale temperature dissipation is extremely rapid and is difficult to capture within the duration between the exhale and its complete dissipation. High-end thermal imaging devices are capable of detecting this rapid dissipation, but are expensive and lack versatility. Therefore, to provide a practical solution for inexpensive thermal cameras, we have devised a *material*-based thermal imaging method for capturing rapidly dissipating exhale behaviors within a medium for respiratory analysis. This material-based method allows us to capture thermal exhale behaviors to solve the rapid dissipation problem and can be used with inexpensive thermal cameras.

3.2 Medium

The performance of this method relies heavily on the choice of medium material based on its thermal retention properties. The ideal medium material should have thermally conductive properties that reflect the temperature changes from the exhale but should also allow for rapid dissipation of the heat between breaths. This material should also be as thin as possible to promote this dissipation process within the material as quickly as possible and should also have a high emissivity so that changes in temperature can be seen with the thermal camera. Therefore the specific material used in the projection of the patient's breath should adhere to a unique set of material properties. It should be (1) thermally conductive, (2) allow for rapid thermal dissipation, (3) be thermally opaque, (4) retain heat signatures long enough to capture, (5) should not introduce material composition patterns into the resulting images, and (6) should be a material that is common and widely available.



(a) (b) (c)

Figure 3: Medium materials (a) cotton, (b) linen, and (c) copy paper with various opaque thermal levels. If the material is not thermally opaque, the patient's face can be seen through the medium, disrupting the quality of our analysis.

In the preliminary experiments conducted with our setup, we explored several medium materials that are commonly available and inexpensive including: stretched cotton, linen, and standard copy paper. The two most important elements that contribute to the effectiveness of the material are defined by (1) the materials thermal dissipation and (2) if the material is thermally opaque. This is because during the experiment the patient is directly on the other side of the camera, making them visible through the medium when they are not breathing. The thermal visualizations of these materials shown in Figure 3 illustrate the problem of seeing the patient's heat signature through the medium for cotton and linen. This problem is solved using a more thermally opaque material: standard copy paper, which has a constant thermal distribution as shown in Figure 3 (c), under the same environmental conditions.

3.3 Embedded Python Processing

For room-temperature operating thermal imaging devices, high-performance is associated with relatively low frame-rates due to the hardware limitations imposed by the internal sensors and computational power of the devices. Similarly, the Raspberry Pi has a limited computational capacity, even for standard imaging and signal processing algorithms on moderately sized datasets. Since we are using Python on embedded systems for device control, image processing, and real-time signal processing, the computational efficiency of the core Python algorithms within NumPy, SciPy, and OpenCV play a critical role within our performance benchmarks. To minimize the impact of these performance metrics, we provide a distributed parallel computation of our core algorithms, and then limit the complexity of the image and signal processing algorithms required to perform the respiratory analysis. This process allows us to port our Python implementation to the compute-bound mobile device and obtain real-time visualizations of the thermal distributions recorded on the projection medium.

3.4 Data Processing and Visualization

The data processing and visualization objective of the prototype system is directed at the implementation of an embedded system running Python that executes on the compute-limited Raspberry Pi. To facilitate high performance within this domain, we emphasize the simplicity of the computational image processing that is required per frame and the input streams from available sensing/-controlled devices. This is due to both the data stream that can be provided from the FLIR C2 sensor, image processing algorithms, and the signal processing required to extract the breathing behaviors recorded on the medium.

The primary processing component of the prototype is based on: (1) thermal sensor image generation, (2) thermal image processing, and (3) the signal processing required to extract the breathing behavior from the set of frames. When a new image is received from the thermal imaging sensor, we apply a two-dimensional gradient-weight function to the image to give a stronger weight to pixels in the center of the image, and less to the pixels near the edge of the image due to the thermal conductivity of the structural frame holding the medium. We subtract the previous image from the current image to get the difference, and then sum the positive values, which indicate increasing temperatures.

$$w(n) = \alpha - \beta \cos\left[\frac{2\pi n}{N-1}\right] \text{ where } \alpha = 0.54,\ \beta = 0.46 \tag{1}$$

To determine the breaths per minute, we multiply the sampled breathing data $b(n)$ by a Hamming window $w(n)$ to define the function $f(n) = b(n) * w(n)$, and then use a fast Fourier transform to convert the data from the time domain to the frequency domain.

$$F(u) = \int_{-\infty}^{\infty} [b(n) * w(n)] e^{-2\pi i n u} dn \tag{2}$$

Based on the implementation of the DFT within NumPy, we expand this to the following expression for the breathing rate within the frequency domain:

$$B(x) = \sum_{n=0}^{N} [b(n) * w(n)] \left[\cos \frac{2\pi x n}{N} - i \sin \frac{2\pi x n}{N} \right] \tag{3}$$

We then take the modulus squared of this convolution within the frequency domain, and find the frequency with the highest amplitude, which indicates the breathing rate in hertz. We multiply this frequency by 60 to get breaths per minute. The following code snippet shows how to accomplish this in Python using NumPy and SciPy. We assume two NumPy arrays are previously defined: data, which holds the thermal sums for each frame, and seconds, which holds the time-stamps for each frame.

```
# Multiply the data by a hamming window
window = scipy.signal.hamming(1000, sym = 0)
data *= window

# FFT transform and modulus squared
fft = numpy.fft.fft(data)
fft = numpy.absolute(fft)
fft = numpy.square(fft)

# Frequency samples
frequencies = numpy.fft.fftfreq(
  data.size,
  d=seconds[1] - seconds[0]
)

# Find the index of the maximum FFT value,
# and get the respiration frequency
max_idx = np.argmax(fft)
breaths_per_sec = frequencies[max_idx]
breaths_per_min = breaths_per_sec * 60
```

Choosing a window size for the thermal data is a delicate balancing act. If the window is too large, the application will not report changes in breathing rate quickly enough. On the contrary, if the window size is too small, the reported breathing rate would have larger fluctuations and a higher error. In our application, we have chosen a window size of 100 data points, and a minimum of 20 data points in the window before starting to calculate breaths per minute. This means that once the window is full, the breathing rate is calculated from the previous 25 seconds of data.

4 EXPERIMENTAL DESIGN

To demonstrate the validity of this method for reliably measuring breathing rate of an individual, we have implemented two different experimental setups using an inexpensive and commercially available thermal camera. In the first experimental design, we provide a proof-of-concept for the medium-based exhale monitoring by using a Raspberry Pi controlled fan. This setup allows for precise control of fan timing, duration, and strength. In our second experimental setup, human subjects are studied to see how our method works under real-world conditions. To visualize these results, we have developed a graphical user interface in Python that displays the thermal data and breathing rate in real-time. This tool can be used by a medical professional to monitor the breathing behaviors of a patient in a natural setting for long-term monitoring.

4.1 Thermal Cameras

We use two thermal cameras in this experiment. The first and primary camera is a FLIR C2 camera. This camera is inexpensive, has an 80x60 thermal sensor array, and streams video to a PC at approximately 5 frames per second. This camera is the primary camera used in our experiments in order to prove that this method works with inexpensive equipment. The second camera used is a FLIR A-series camera with a custom filter that allows us to view exhaled CO_2 gas directly. This camera has frame rates of 30 and 60[fps], has a 640x512 thermal sensor array, and is relatively expensive. The A-series camera is used for verifying different medium materials, and is used as a ground-truth reference for data taken with the C2 camera for the medium.

4.2 Fan Experiment

The fan experiment is designed to simulate realistic breathing activity, and gives us the ability to programmatically set variables that are difficult to control in human subjects, such as breathing rate and strength. In this experiment, a fan blows air through a heated brass tube onto the surface of the medium, assembled by 3D printed components.

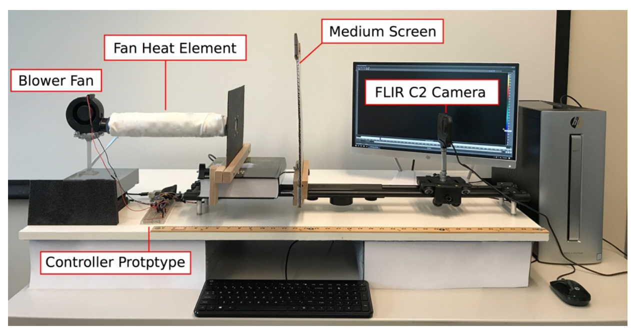


Figure 4: Fan experiment setup illustrating the overall setup of the fan, medium, and thermal camera. The heating element provides a constant heat source that simulates human breathing through the brass tube. The medium separates the exhale and the thermal camera.

The brass tube is heated to approximately 60℃ so that air passing through the tube will be heated to a temperature similar to that of human breath. The heated air leaves a heat signature on the surface of the medium, and the C2 camera records the heat signature on the opposite side. Figure 4 shows our experimental setup.

This experiment setup uses an off-the-shelf squirrel cage blower controlled by a Raspberry Pi 3 model B. The Raspberry Pi acts as a variable power supply to the fan by setting a digital potentiometer that limits the output voltage. The Raspberry Pi runs a Python script at startup that uses the PySerial module to listen for incoming serial commands, and sets the fan voltage with the Spidev module. In this way, the fan and the thermal camera are controlled programmatically, and can therefore be synchronized. The image and diagram within Figure 5 shows our Raspberry Pi fan controller prototype (top) and the circuit diagram (bottom) used to implement the simulated exhale controller.

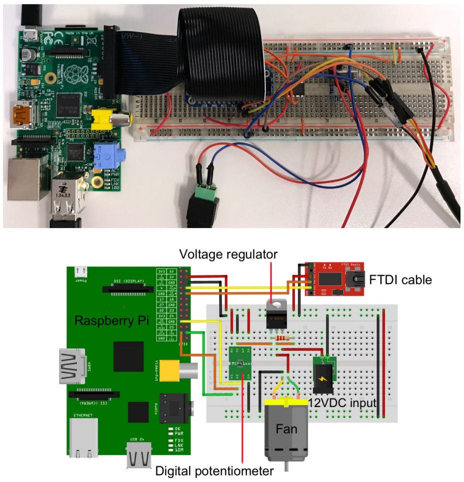


Figure 5: Raspberry Pi automated fan controller prototype. The implementation (top) provides the fan controller design (bottom) for controlling the fan and performing computations for breathing analysis.

The fan experiment setup lets us set a constant known breathing rate with which to compare to the thermal signal, and it lets us set a consistent air speed and location of the thermal signature on the medium. This is important for evaluating both the medium thermal material properties as well as the exhale heat propagation. In each of our fan experiments, we set the rate to 10 BPM, and limit the fan speed to the lowest possible voltage, which is approximately 5V. Images obtained from this experiment show a central, round gradient that appears and grows when the fan is turned on, and shrinks and disappears once the heat source is removed. Figure 6 shows examples of thermal images obtained from this experiment using the fan controller.

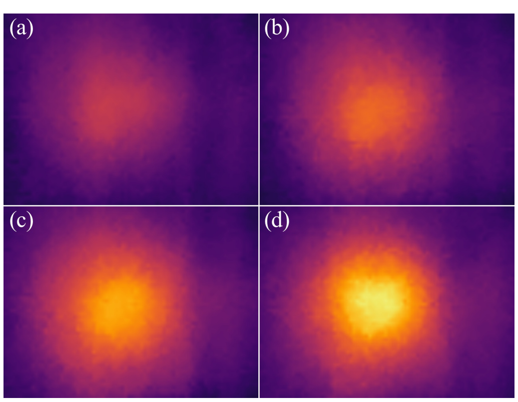


Figure 6: Fan simulated exhale results. The images show a clear thermal distribution on the medium tied to the simulated exhale that corresponds to both duration and strength.

When recording simulated exhale data, we record the on and off times of the fan, as well as the the thermal images from the camera. Figure 7 shows the thermal sums from the fan-based simulated exhale plotted against the on and off times of the fan. The results show a strong correlation between the thermal signal and the rate at which the fan is turned on and off.

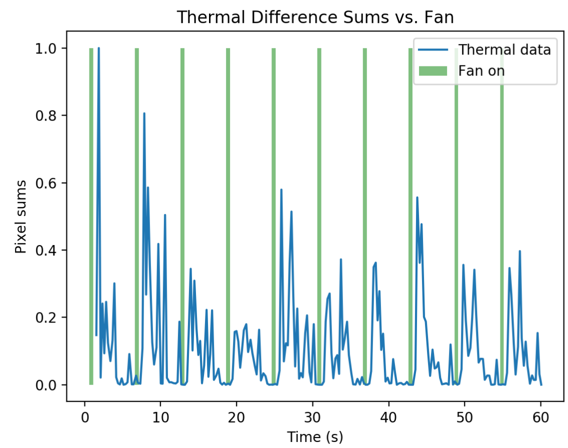


Figure 7: Thermal sums versus fan on/off timing. This indicates a strong correlation between the simualted exhale temperature on the medium and the ground-truth timing of the fan control.

4.3 Human Experiment

The human experiment examines natural breathing from human subjects. In this setup, the C2 camera and the medium are both mounted on a tripod. The subject sits in a chair, and the medium and camera are positioned so that the exhaled breath from the subject collides with the medium. As a way of validating the heat signatures on the medium, the A-series camera records the exhaled breath simultaneously from the side. From this data, we know exactly when each breath occurs, as well as whether the source of the exhale was the mouth, nose, or both. Figure 8 shows our experimental setup.

Figure 8: Human trial experimental setup. The C2 records the medium thermal signature and the A-series camera provides the ground-truth values for the breathing rate and duration.

Images from the A-series camera that visualizes CO_2 shows a side profile of the patient's face, as well as a jet stream of air during normal exhalation. Figure 9 shows examples of images obtained from the A-series camera, demonstrating no exhale (a), exhale from the mouth only (b), exhale from the nose (c), and simultaneous exhale from the nose and mouth (d). The duration and timing of these recorded exhale behaviors are then compared with the medium-based trial to verify that the exhale behaviors extracted from the medium match the actual behaviors.

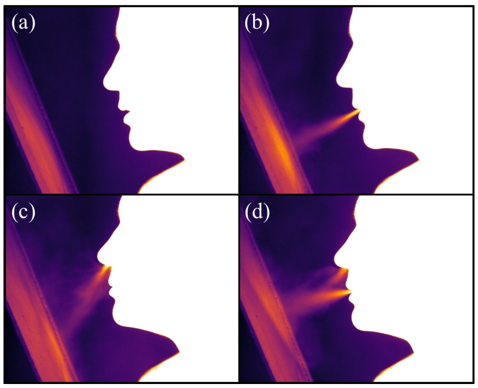


Figure 9: Images from the A-series CO_2 camera showing (a) no exhale, (b) exhale from the mouth, (c) exhale from the nose, and (d) exhale from the mouth and nose.

Subjects are asked to alternate breathing between their nose, mouth, or both throughout the course of each 60 second trial. We have developed a graphical user interface to display results from these trials, providing an interactive form of monitoring the real-time data generated by our approach. The GUI is written using PyQt5, and plots data using PyQtGraph. We continuously cycle data from these 60 second trials to analyze real-time data from a camera. The application calculates the positive difference sums of each image and stores them in a window queue. Once this window contains enough data, the application performs an FFT on the entire queue when a new frame is received. Once the queue is full, old data is removed to resume the moving window analysis. The window holds approximately 25 seconds of data. Figure 10 shows our GUI processing information from the fan experiment.

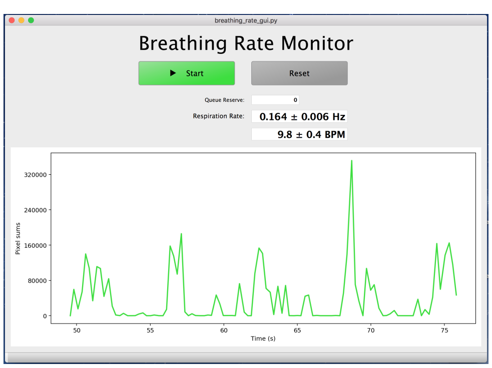


Figure 10: Breathing rate monitor GUI.

5 RESULTS

The results generated by using our prototype experiments are evaluated by two criteria: (1) the results of the experiments with respect to breathing rate and (2) the results of measuring breathing strength. In addition to these two results we also analyze the human-based trials by evaluating the nose/mouth distribution using the medium.

5.1 Respiration Rate

In our results we simply classify the breathing rate results from the fan-based simulated exhale and the human-based experiments as breaths per minute (BPM). This value is indicated in the interactive application as the respiration rate, which is updated in real-time.

5.1.1 Fan Experimental Results. To determine the accuracy of our approach, we analyze the results from the fan-based experiment as our control. Since we are able to programmatically set the breathing rate in this experiment, we have a set internal and ground-truth with which to compare results from our prototype. In the fan experiment setup, the end of the brass tube is located six inches away from the medium, and the C2 camera is placed 15.5" away from medium in order to record as much of the surface area of the frame as possible. The medium is a 11"x17" piece of copy paper. We conduct three 60[s] trials with this setup. Each of these data sets are processed, and breaths per minute statistics are recorded for each individual experiment. Table 12 shows the statistics of the calculated breaths per minute (BPM) for each trial.

Fan Simulated Breaths Per Minute		
Sample	Mean	Std. Dev.
1	10.00	0.203
2	9.996	0.159
3	10.008	0.160

Figure 11: Resulting breaths-per-minute statistics from three fan trials. This demonstrates the validity of our approach under ideal circumstances.

The minimum breaths per minute from sample 1 has the greatest absolute difference from the actual rate of 10 breaths per minute at 9.57 breaths per minute. This is an accuracy of ± 4.3% in the worst case, giving us confidence that our processing method is accurate under ideal conditions imposed by the fan-based experiment. This validates our initial hypothesis that our method can be used to accurately track exhale rate through the medium. However, in order to assess the robustness of this method, we must look at the results of the human experiment.

5.1.2 Human Experimental Results. In the human-based form of the experimental setup, numerous additional factors contribute to the degradation of the breathing signal that we record on the medium. These include: (1) the patient's distance to the medium, (2) the orientation of their face and exhale, and (3) the natural breathing patterns incurred by exhaling through the mouth rather than through a symmetrical tube which generates a much more turbulent flow. Based on these factors, we see the naturally occurring reduction in the accuracy of the rate, however this reduction in accuracy remains relatively small throughout our trials.

Human Breaths Per Minute from Medium and Reference				
Sample	Medium Mean	Side Mean	Medium Std. Dev.	Side Std. Dev.
1	17.455	17.581	1.346	1.205
2	10.509	10.801	0.471	0.270
3	10.577	10.299	0.398	0.294

Figure 12: Breaths-per-minute statistics from three human trials, comparing medium data and side-view data.

For the human experiment, we collect 60 second samples of data from three different subjects breathing onto the medium. As expected, these samples are far more chaotic than the thermal signatures from the fan experiment, in breathing rate, location of the thermal signature, and shape of the thermal signature on the medium. As previously mentioned, we simultaneously record thermal data with the A-series camera from a side view to validate the breathing rate from the medium data. We process the reference data in the same way that we process the medium data, but we crop out the right half of the image to remove the face of the subject, leaving only the exhaled air and the medium visible. We calculate the breaths per minute of both the reference data and medium data for 60 seconds, and then compare these data sets. Figure 13 shows the breaths per minute from each data set plotted on each axis. A clear linear correlation exists between the two data sets.

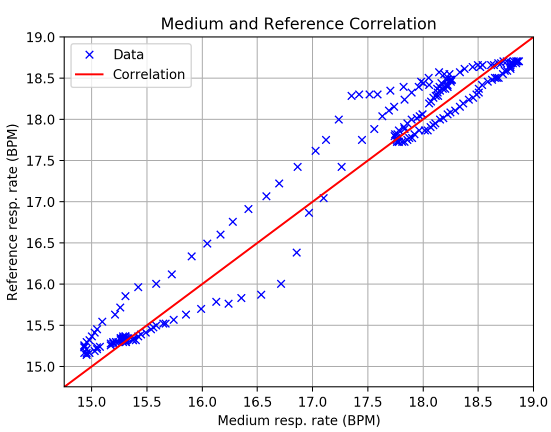


Figure 13: Correlation between the reference and medium data from our human subject from our experiments.

From this data, we can see that the subject is altering their respiration rate between 15 and 18[BPM] over time. Figure 14 shows another view of this data as a histogram, showing the strong correlation between the reference and medium evaluation.

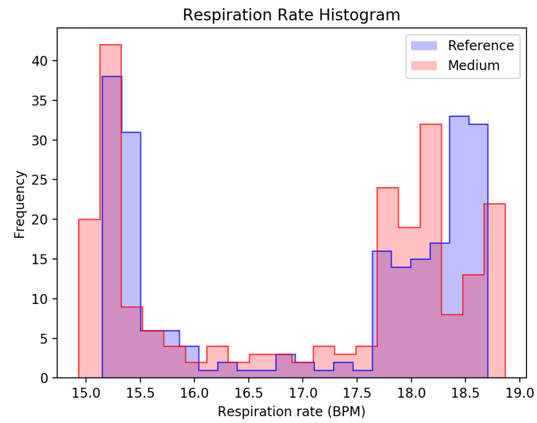


Figure 14: Histogram of reference and medium data from our human subject, showing the strong correlation between the reference and medium.

5.2 Breathing Strength

Another metric that is potentially attainable through this method is breathing strength. If the subject changes their breathing strength over time, this should theoretically result in a visible difference in the thermal signature on the medium, and in the resulting difference sums. To test this theory, we use the fan experiment setup to simulate different breathing strengths. This is done by changing the voltage supplied to the fan, which modifies the speed of the fan. In this experiment, we simulate breathing at 10 BPM, and toggle between a light and forceful speed every other breath. Figure 15 shows the resulting data from this experiment. The results show a clear change in the amplitude of the waveform between breaths.

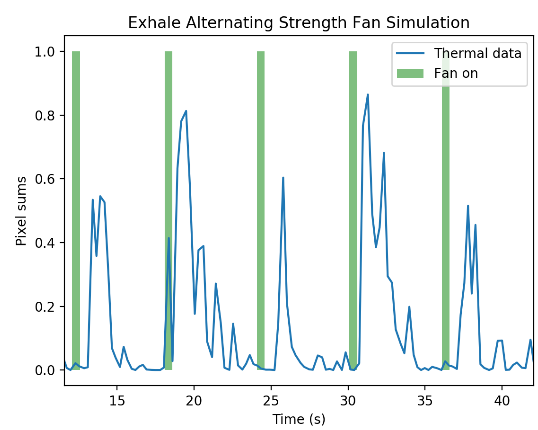


Figure 15: Simulation of light and forceful exhales with the fan experiment. This experiment illustrates that we can detect varying strength exhales based on the thermal signature on the medium.

In human trials, participants are asked to alternate between a light exhale and a more forceful exhale with each breath. Figure 16 shows a plot of the increasing difference sums over time from this experiment. There's clear separation in amplitude between the light exhales and the forceful exhales, meaning that the exhale strength of an individual is a metric that can be extracted from the thermal data that we collect on the medium.

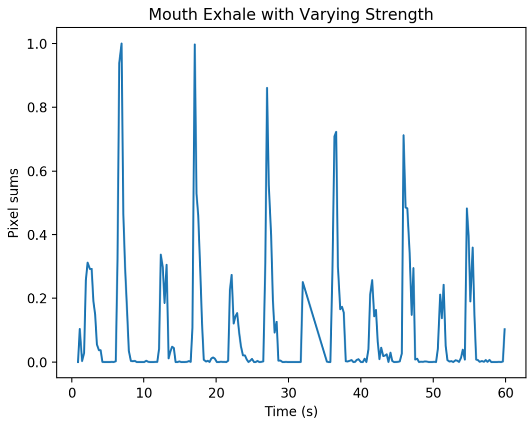


Figure 16: Alternating light and forceful exhales from a human subject. The clear separation in the magnitude of the thermal signature allows us to provide a breathing metric for breathing strength.

5.3 Nose and Mouth Separation

In human trials, participants are asked to breathe from their nose only, mouth only, and then to breathe naturally. In these trials, the thermal signature on the medium differs depending on how the subject breathes. Figure 17 shows a thermal image of the medium during mouth breathing on the left, and a thermal image of nose breathing on the right. These images differ in the shape and location of the thermal signature on the medium. It is feasible that the distribution of airflow between the nose and mouth can be extracted from these thermal signatures, and used to determine if the subject is having difficulty breathing through their nose.

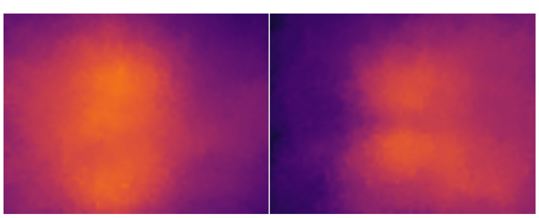


Figure 17: Exhale from mouth (left) and exhale from nose (right) from a rotated view. Exhale signatures between each pattern are clearly visible.

5.4 Prototype Performance Analysis

To evaluate the performance of our software on a low-powered multi-core CPU, we use a Raspberry Pi (3) model (B) to process thermal data taken with the C2 camera. Our optimized algorithm processes frames at a much faster rate than the room-temperature thermal camera, therefore this indicates that images from this particular device could be processed with less powerful hardware. However, our system could be used with a thermal camera with a higher frame rate and larger image size, which may present a challenge to low-power hardware. To determine the our system's capabilities when used with more powerful thermal cameras, we change the thermal image data-stream between 160x120, 320x240 and 640x480 resolution images and measure the average processing time per image. If the frame rate from the camera is too high to process in real time, we are forced to either drop frames to maintain real time processing, or we must parallelize the processing function and add buffering to process multiple buffered frames at once. The diagram in Figure 18 illustrates how we compute the difference images in parallel using multiple cores.

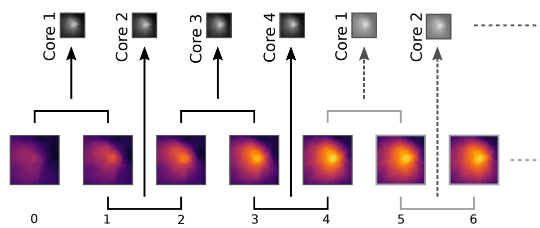


Figure 18: Parallel distribution of the core executions for computing the difference images of the video stream.

To determine the performance of parallel processing at high frame rates, we store the frames in a buffer and use the pathos multiprocessing module to process multiple frames simultaneously. We chose to parallelize the image processing steps that take the most time and are not order-dependent: differencing two images, blurring the resulting difference images (for noise tolerance), and summing the computed values. Our implementation is described as follows: two images are passed to each core, and each core processes the difference between the two images, blurs the difference image, and sums the positive values in the difference image, as shown in Figure 18. We demonstrate our results from running our program with different image sizes and cores as shown in Figure 19.

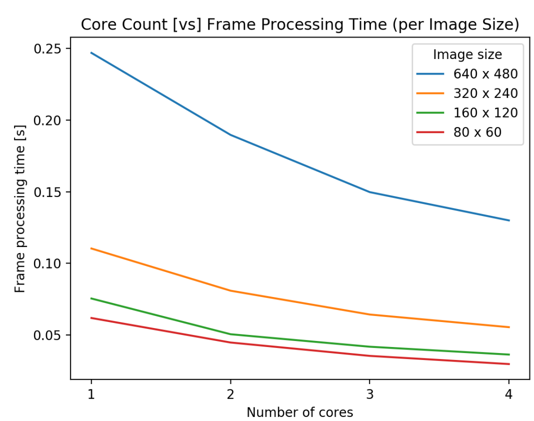


Figure 19: Comparison of processing of various image sizes computed in parallel using 1 to 4 cores on the Raspberry Pi (3). Our parallel implementation using the pathos multiprocessing module allows real-time (respiratory-rate) monitoring through a self-contained embedded system.

6 EVALUATION AND DISCUSSION

A small, independent embedded device that monitors respiratory activity would be an effective and versatile tool in clinical pulmonary and sleep-related studies where comfort of the patient is a primary objective. Our results show that we can make accurate predictions about respiratory behaviors in an environment that is natural and comfortable for the patient, and that inexpensive low-powered hardware is capable of processing common thermal image sizes at reasonable rates. Python's extensive scientific computing modules, as well as easy-to-use image and parallel processing modules, provide the possibility of extending this software to include additional respiratory analysis, such as measuring exhale distribution, strength, and tidal volume estimates. Through our real-time processing, we have begun to evaluate use-cases and clinical implications of using a self-contained respiratory monitoring devices for short-term studies related to natural breathing behaviors. To further evaluate the impact of this design, we also consider the medical significance of our method and analyze the challenges associated with iterating the development of our working prototype into clinical experiments to determine the feasibility of the proposed system.

6.1 Medical Significance

This method of respiration monitoring has many implications for the medical field for breathing behavior monitoring and evaluation. This method is ideal for sleep studies and provides precise monitoring of natural and unhindered breathing behavior, as it measures respiration directly, but allows the patient to breathe without obstruction or discomfort. Because this method works regardless of the age and physical characteristics of the individual, it could be used to monitor the respiration of small children or infants. This method has the potential to be used to measure other respiratory metrics as well, such as breathing strength, tidal volume, and nose to mouth distribution. Nose to mouth distribution is a valuable metric for medical professionals who need to identify when a patient is having trouble breathing through their nose, which may indicate some form of respiratory obstruction. This could be particularly important for determining whether an infant is having difficulty breathing through their nose for extended durations, which can be detrimental to their physical development.

6.2 Challenges

One fundamental challenge that needs to be addressed to make this method robust is how to handle patient movement during sleep-based studies. The medium needs to be in line with the subject's face despite how the patient moves their head, and the camera needs to be able to record the thermal signature where the camera image plane is parallel to the medium. Another significant challenge is determining the optimal distance to place the medium from the patient. If the medium is placed too close to the patients face, they may become uncomfortable and tend to look away from the medium. If the medium is too far away, the breathing behavior becomes less prominent and the respiratory behaviors are harder to identify and extract as unique signatures. Both of these factors contribute to difficulties in obtaining reliable respiratory behaviors over an extended period of time.

7 CONCLUSION

In this paper, we have described a method of non-contact respiration rate monitoring that provides a more direct measurement than its counterparts. We have introduced a graphical user interface for use by medical professionals that displays the thermal data and breathing rate of the patient in real time. Results from our fan experiment show that we are able to extract breathing rate from this method with reasonable accuracy. We have also demonstrated the potential for other vitality metrics such as breathing strength and nose to mouth distribution to be extracted from the thermal signature. The combination of Python processing and a low-powered CPU allows this system to be a small, standalone device, while still processing thermal data in real time. This method is suitable for a wide variety of patients, and can be implemented as a low-cost medical device on minimal hardware.

8 ACKNOWLEDGMENT

This work was partially funded by DoEd GAANN Fellowship: P200A150283 and NSF Grant: 1602428.

REFERENCES

[1] Abdulkadir Hamidu Alkali et al. 2013. Facial tracking in thermal images for real-time noncontact respiration rate monitoring. In *Modelling Symposium (EMS), 2013 European.* IEEE, 265–270.
[2] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's J. of Software Tools* (2000).
[3] RJ Canter. 1986. A non-invasive method of demonstrating the nasal cycle using flexible liquid crystal thermography. *Clinical Otolaryngology* (1986), 329–336.
[4] Mathias Cehlin, B Moshfegh, and M Sandberg. 2002. Measurements of air temperatures close to a low-velocity diffuser in displacement ventilation using an infrared camera. *Energy and Buildings* 34, 7 (2002), 687–698.
[5] A. R. Fekr, K. Radecka, and Z. Zilic. 2015. Design and Evaluation of an Intelligent Remote Tidal Volume Variability Monitoring System in E-Health Applications. *IEEE Journal of Biomedical and Health Informatics* 19, 5 (Sept 2015), 1532–1548.
[6] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001. SciPy: Open source scientific tools for Python. (2001). http://www.scipy.org/
[7] Gregory F Lewis et al. 2011. A novel method for extracting respiration rate and relative tidal volume from infrared thermography. *Psychophysiology* 48, 7 (2011), 877–887.
[8] Wansuree Massagram, Victor M Lubecke, and Olga Boric-Lubecke. 2009. Microwave non-invasive sensing of respiratory tidal volume. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE.* IEEE, 4832–4835.
[9] George B Moody et al. 1986. Clinical validation of the ECG-derived respiration (EDR) technique. *Group* 1, 3 (1986).
[10] Kundan Nepal, Eric Biegeleisen, and Taikang Ning. 2002. Apnea detection and respiration rate estimation through parametric modelling. In *Bioengineering Conference, 2002. Proceedings of the IEEE 28th Annual Northeast.* IEEE, 277–278.
[11] Cheng-Li Que et al. 2002. Phonospirometry for noninvasive measurement of ventilation: methodology and preliminary results. *Journal of Applied Physiology* 93, 4 (2002), 1515–1526.
[12] Karl Storck, Matts Karlsson, Per Ask, and Dan Loyd. 1996. Heat transfer evaluation of the nasal thermistor technique. *IEEE Transactions on Biomedical Engineering* 43, 12 (1996), 1187–1191.
[13] K Song Tan et al. 2010. Real-time vision based respiration monitoring system. In *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on.* IEEE, 770–774.
[14] Shane Transue, Phuc Nguyen, Tam Vu, and Min-Hyung Choi. 2016. Real-time tidal volume estimation using iso-surface reconstruction. In *Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016 IEEE First International Conference on.* IEEE, 209–218.
[15] Stefan van der Walt, S. Chris Colbert, and Gael Varoquaux. 2011. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering* 13, 2 (2011), 22–30. https://doi.org/10.1109/MCSE.2011.37 arXiv:http://aip.scitation.org/doi/pdf/10.1109/MCSE.2011.37
[16] Z Wang, PT Ireland, and ST Kohler. 1998. Gas temperature measurement in internal cooling passages. *ROLLS ROYCE PLC-REPORT-PNR* (1998).
[17] Zhen Zhu, Jin Fei, and Ioannis Pavlidis. 2005. Tracking human breath in infrared imaging. In *Bioinformatics and Bioengineering, 2005. BIBE 2005. Fifth IEEE Symposium on.* IEEE, 227–231.