Wildfire Detection Testing for UAVs and CubeSats

Trinity Barnett*, Evan Baxter†, Alvin Cortez‡, Allison White§
Mississippi State University, Mississippi State, MS 39762

The purpose of this project was to use the near-infrared spectrum of light to identify potassium spikes from burning vegetation. The imaging system using a Raspberry Pi 4 Model B and Raspberry Pi High-Quality Camera along with bandpass filters can be effectively used to locate burning biomass. Camera system gains, ISO, and shutter speed were adjusted to minimize pixel saturation. Images of flames from biomass fires were post processed to isolate wavelengths of interest, specifically potassium emissions. This was shown to be an effective means of detecting burning vegetation. For physical stabilization, a ground-testing camera mount was created for the camera and filters. A filter holder was developed to attach to the camera mount to ensure the filters could be swapped by means of a servo. Subsequently, a bus was developed for the compact camera payload as well as the other system components. The enclosure will then be adapted to attach to a drone. This allows for overhead fire image capturing and testing from further distances. The result was a camera system that takes quality images, switches filters automatically, and utilizes three separate bandpass filters. The system can be modified for further UAV or cubesat applications.

Nomenclature

λ	=	wavelength
SS	=	shutter speed
ISO	=	sensitivity to light
K	=	potassium
NaCl	=	sodium chloride
O_2	=	oxygen
IO	=	input/output
s76	=	array of sum of pixel values from an image using 760 nm lens filter
s77	=	array of sum of pixel values from an image using 770 nm lens filter
s78	=	array of sum of pixel values from an image using 780 nm lens filter

I. Introduction

THE unpredictable nature of wildfires demands early identification and analysis to prevent disasters. There is a demand for unmanned aerial vehicles (UAV) as well as cubesats to recognize wildfires, so human lives will not be put at risk. Long term, UAVs and cubesats are more cost effective than past methods of collecting similar data. There are a few ways to detect a fire in this manner. For the purposes of this project, wavelengths of light on the near-infrared spectrum are being analyzed. To separate wildfires from other light sources, the fuel source for the fire must be considered. Wildfires typically burn organic matter. When vegetation is burning, there is a spike in the spectral radiance at a particular wavelength as seen in Fig. 1. Spectral radiance describes how energy in an

* Undergraduate Student, Aerospace Engineering, AIAA Student Member.
† Undergraduate Student, Aerospace Engineering, AIAA Student Member.
‡ Undergraduate Student, Aerospace Engineering, AIAA Student Member.
§ Undergraduate Student, Aerospace Engineering, AIAA Student Member.

electromagnetic emission, like light from a fire, is distributed over wavelength, λ. The λ where this spike is located is between 760 nm and 780 nm. This sudden increase of spectral radiance within the λ bounds indicates there is potassium being burned. When potassium spikes, it indicates the fire is fueled by organic matter; therefore, a wildfire is identified.

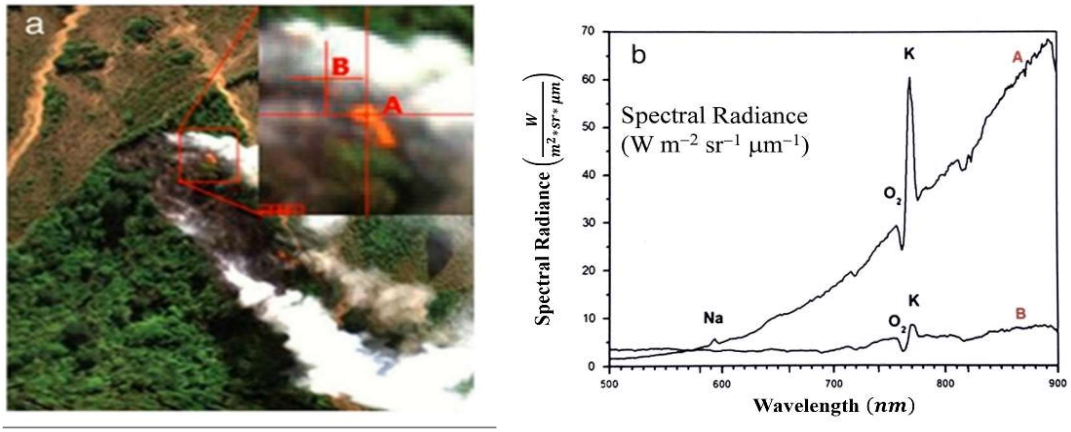


Fig. 1 Spectral radiance over wavelength plot of wildfire taken with hyperspectral instrument from Ref. 1.

To find this λ of light, a system must be designed to block out all other wavelengths. The solution to this problem is to use different filters. There are three filters from Thorlabs used for the system. Thorlabs models FBH760-10 [2], FBH770-10 [3] and FBH780-10 [4]. Each bandpass filter allows specific wavelengths of light into the lens of the camera. For wildfire applications, 760 nm, 770 nm, and 780 nm filters [2,3,4] are being used. Figure 2 illustrates the percent transmittance of these wavelengths and their tolerances. They block out all wavelengths outside of the ones shown. A mechanical system is being designed to switch filters autonomously.

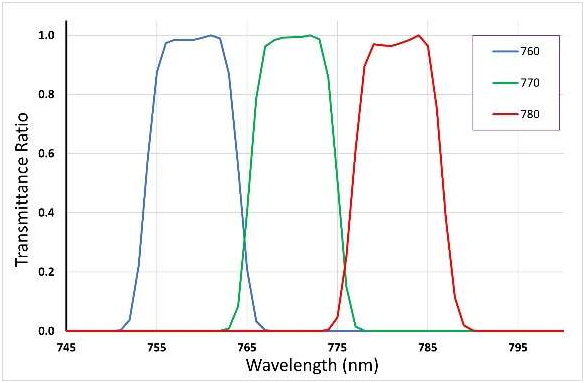


Fig. 2 Transmittance at wavelengths for the 760 nm, 770 nm, and 780 nm filters [2,3,4], respectively.

The main component of the wildfire detection system is a Raspberry Pi HQ Camera [5] which has been modified for near infrared applications. It has the ability to be focused and programmed using Python [6] scripts. The camera [5] settings deemed acceptable are those in which none of the image pixels are saturated. Tests were performed using candles, wooden dowels, pine needles, paper, and cardboard. Candles and wooden dowels were used to optimize code and get baseline data from flames. The pine needles simulate a wildfire since most wildfires contain burning pine needles. Paper and cardboard results were also compared to the pine needles.

To get consistent images during testing, a camera mount was designed in SOLIDWORKS [7]. It also allows for filters to be switched easily and moved without disturbing the setup. The ground-testing filter holder has slots for four filters while one remains empty. A bus was designed to carry a newer filter system. This new system includes a camera holder, and a filter holder which rotates by means of a servo [8]. There are three slots for three filters on this payload design. The bus will also contain the necessary components to make the system operate from a UAV.

II. Camera System Development

A. Initial Ground-Test Camera Design

To view a fire within the near infrared spectrum, the filters that normally block infrared light on the Raspberry Pi High Quality Camera lens were removed. The camera was then controlled by a Raspberry Pi 4 Model B computer [9] connected via ribbon cable. The entire system was powered by an Anker battery [10]. The Pi [9] uses Raspberry Pi OS which serves as a platform to store imaging programs. All images were taken using these programs written in Python [6]. Since the spike of light emitted from burning potassium occurs at a wavelength of 768 nm, the camera was used with 770 nm and 780 nm filters [3,4] as well as with no filters. The Pi [9] was controlled directly with a keyboard and mouse. A flash drive was used to transport photos from the Pi [9] and onto a PC for image processing. Figure 3 depicts the camera system equipment.

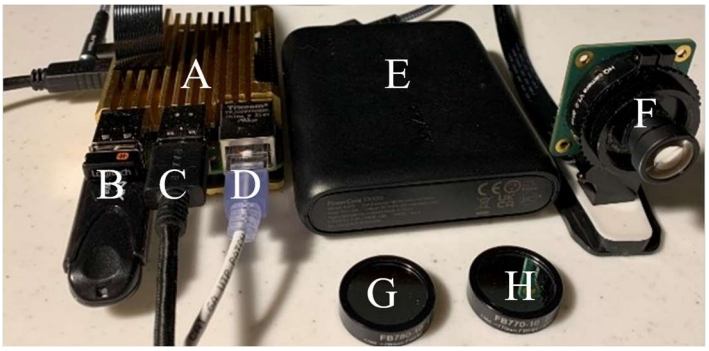


Fig. 3 Raspberry Pi camera setup and filters. *A) Pi [9]. (B) USB storage and mouse. (C) Keybord. (D) Ethernet cable. (E) Battery [10]. (F) Camera [5]. (G) 780 [4] nm filter. (H) 770 [3] nm filter.*

Before testing the camera [5] with fires generated from wood, the camera [5] had to be adjusted such that the pixels would not saturate. This was achieved by using an unscented candle as a reference flame. The initial images of the unscented candle used varied the gain values of the camera [5] to produce different colors and saturations. Camera gains are coefficients that factor into how the camera translates raw image data into the visible spectrum of light. This creates a unique situation where the camera absorbs near-infrared data but processes the data as if it is visible. To counter this, the red, green, and blue (RGB) values of each pixel in the images were used in a MATLAB [11] script to form histograms. The histograms of the separate RGB values ensure each value remained as similar to the others as possible with no extreme variation. This would allow for a more accurate prediction of pixel saturation as well as easier saturation reduction. Saturation is defined as the maximum RGB pixel value of 255. It was found that gains of 1.0 and 1.0 (Appendix A) produced almost greyscale images that had close to equivalent histograms for each value. This trend continued with and without each filter. The initial set of images testing the 1.0 and 1.0 gains had an error in which the background of the candle was reflective. From this point forward, a blank background was used.

The final step was to reduce saturation as much as possible. Images with the highest saturation in the prior test were taken without either filter. Thus, images with no filter were used to determine the optimal camera settings. The two variable settings were the shutter speed, SS, and ISO sensitivity. First, a series of images were captured with decreasing shutter speed starting at 2000. The ISO was kept at the standard 200. The images were then input into a MATLAB [11] script that counted saturated pixels, and a trend of decreasing saturation as SS decreased was identified. Table 1 displays these trends. The lowest impactful value for the shutter speed was estimated to be around 200. The ISO was then manipulated for further saturation decrease. This time, the SS was constant at 200. The low value of 50 ISO provided a mostly unsaturated image. Thus, the shutter speed was kept at 200. However, the ISO remained constant at 200 for the vegetation-burning tests.

Table 1. SS in relation to saturated pixel using constant ISO of 200.

Shutter Speed (milliseconds)	Number of Saturated Pixels (out of 12E6)
2000	30261
1000	24081
500	22152
250	19843
200	19482
150	18838
100	19393
50	18759

B. Final Camera Design

The final camera [5] design was focused on the preparation of the camera [5] as a flight payload for a UAV or cube satellite. This implies a need for autonomous filter switching and image capturing. To switch filters [2,3,4], a servo [8] assembly was calibrated such that the center of the filter [2,3,4] would be aligned with the camera [5] lens when rotated. The image capturing was then timed such that a burst of images would be taken for each filter. The filters included the original 770 [4] and 780 [5] nm wavelengths in addition to a 760 [2] nm wavelength filter, so the servo [8] had to accommodate alternating between three filters in total.

First, the Raspberry Pi High Quality Camera [5] was sent to Innovative Imaging and Research Corporation (I2R) for calibration. This reduced the RGB value fluctuation even further and provided a near identical match of the RGB values. The camera code was also updated to the picamera2 python library for efficiency of capturing and variety of capturing options. The settings determined in the initial phase remained the same. Then, a user-friendly image burst code using this library was written to accelerate ground testing. The program allowed users to save bursts of images in larger quantities as a controlled fire developed. The image burst code used within this ground test model directly translated into the servo integration.

The servo itself was a TowerPro SG-5010 [8] connected to the Raspberry Pi 4 Model B [9] via IO pin 18 and controlled using the standard gpiozero python library. The connection of the Raspberry Pi High Quality Camera [5] and servo [8] can be seen in Fig. 4. Before implementation, the pulse width of the servo [8] was altered such that a smooth 91.4-degree rotation could be achieved. Due to the space requirements of the filter holder, the center of each filter [2,3,4] is 45.7 degrees apart. Once smooth rotation was reached, a short program that rotated the servo [8] in increments of 45.7 degrees (cycling through each filter and back) was run to observe the error of the servo over time.

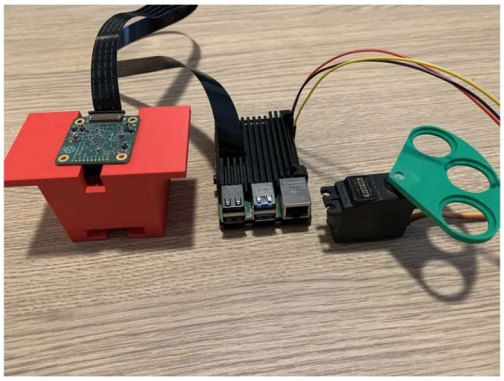


Fig. 4 Raspberry Pi camera [5] connected to Raspberry Pi 4 [9] via ribbon cable with TowerPro servo [8] connected via IO pin 18. Camera is held in place by ground test model, and servo supports filter holder.

The final test of the camera [5] assembly was to coordinate the servo [8] and the image burst programs. This was done by simply merging the prior two scripts. The result was a camera system that can capture three images with separate filters [2,3,4] in under 5 seconds.

III. Structural Development

A. Ground-Based Design Phase

To properly operate the camera, a suitable mount had to be made for it. Since the project is a collaboration with Meridian Community College (MCC), models were made for their purpose as well. All the modelling done for the camera [5] and lens filters was done in SOLIDWORKS [7]. To align with the plans for initial testing, a camera holder with the purpose of ground testing was made. Figure 4 shows the initial model for the ground-based design. The purpose of the flanges is to function as clamping surface so that the camera [5] can be sustained above a fire; this design fits well within the requirements of MCC, as they do their testing in a lab with the camera [5] in an overhead position. Another significant part of the system was the holder for the lens filters. The design that was modeled and used had four slots for lens filters since MCC used four filters for their project. No adjustments needed to be made since the wildfire detection system used three lens filters [2,3,4]. It consisted of a top and bottom holder. The holders included standoffs to hold the lens filters, and a fitting mechanism was introduced to secure everything together. Despite thorough design considerations and modelling, a camera mount system from a previous team was used. The reason for this was the lack of manufacturing capabilities.

The majority of the difficulties for the structural development resulted from the absence of manufacturing capabilities. Several 3D printing attempts were made with the available printer, a Creality Ender 3 Pro, to no success. The prints would typically come out with bowed edges, meaning some sides of the print would have seemingly bent surfaces. Additionally, the unknown tolerance of the printer made it difficult to print holders that would fit together. The printer would also fail frequently, making it nearly impossible to perform any test prints. To combat this problem, new options were explored. Fortunately, connections with the Advanced Composites Institute (ACI) were solidified and parts could be printed quickly and accurately. ACI provided three resources to the team: design planning, manufacturing, and structural analysis assistance. Parts were then rapidly produced for both MCC and the wildfire detection payload. Figure 5 shows the final design of the camera holder and filter holder. The holes in combination with toothpicks or pins provided more stability; the new fitting mechanism was a major improvement as well, as the last design was very brittle and would break easily.

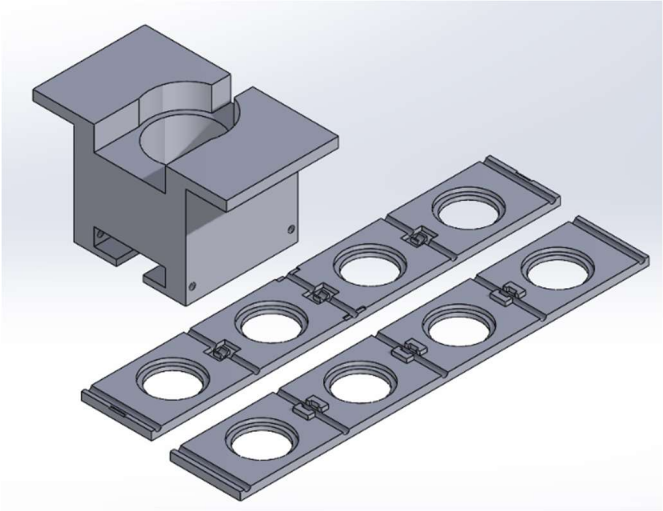


Fig. 5 Final Model for Camera Mount and Filter Holders

B. Autonomous System for Flight Applications

The next phase of the project aimed toward creating an autonomous system for UAV and cubesat implementation. The initial design for the autonomous system consisted of one large gear, a small gear, a custom shaft, a circular filter holder, and a 180-degree sweeping servo [8] that would fit within a 1.5U cube satellite. For context, a U represents a 10 cm x 10 cm x 10 cm cube with the length being 15cm in this case. The gear ratio was selected to be 2:1 with the larger gear having 24 teeth making the smaller gear hold 12 teeth; their respective radii were 50 mm and 25 mm. The gears were designed in SOLIDWORKS [7] with the selected number of teeth allowing 360-degree rotation in fifteen increments. Afterwards, a cross-shaped shaft with a 256 mm^2 cross-sectional area was designed to be the center of rotation for the large gear and filter holders.

The filter holder consisted of two identical circular filter holders. The filter holder for this system contained four evenly spaced slots that would hold the 760 [2] nm, 770 [3] nm, and 780 [4] nm filters along with an additional slot for MCC's purposes. Each half of the holder had a radius of 55 mm with a thickness of 4 mm; the cuts made for the filters were 2 mm deep with 2 mm thick standoffs. The final part of the system included a simple redesign of the camera holder. The flanges were removed since it was designed to be held in an enclosure, and a cut was made to allow the filter holder to rotate underneath. Once all the parts of the system were modeled, an assembly, shown in Fig. 6, was created in SOLIDWORKS [7] to obtain an accurate arrangement of the system; a dynamic simulation was then performed to ensure the system's functionality.

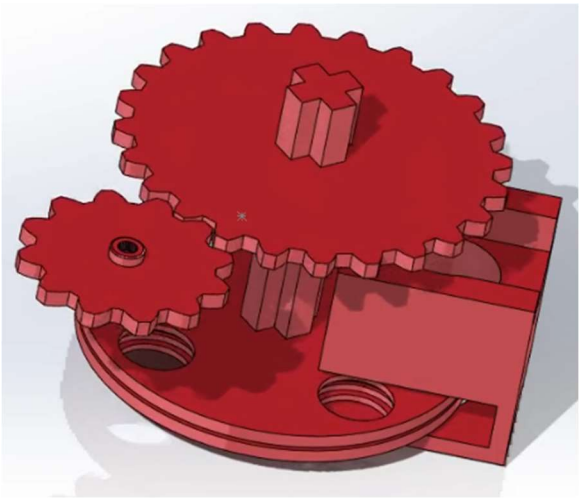


Fig. 6 First design for rotating filter holder assembly

The manufacturing of these parts was done through ACI. Thanks to their high-end printing capabilities, the prints were clean, and the large gear and filter holders fit perfectly on the shaft. The smaller gear, however, was designed to fit perfectly on the shaft of the servo [8]. Unfortunately, the grooves on the servo's shaft were as small as 0.14 mm, and the printer's maximum precision was $\pm$ 0.2 mm. A temporary solution was found by fixing the gear to a pre-existing attachment that was included in the servo's product package. The system worked as intended, but it proved to be too large for the desired 1.5U cubesat constraint. This problem called for an immediate redesign to downsize the system. The new approach involved a new filter holder as shown in Fig.7 that would be directly attached to the servo shaft.

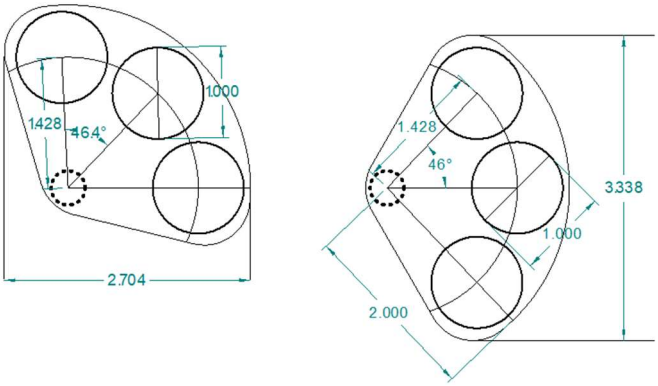


Fig. 7 Sketch for new filter holding method.

This significantly reduced the size of the system by eliminating the need for a shaft and gears. The new filter holder was then printed with our new Bambu X1 printer. Since the printing precision was limited to 0.2 mm, a small hole was made to fit tightly onto the servo [8]. Several tests were run afterwards to observe any slipping that might occur between the shaft ridges and smooth surface connecting the holder and servo [8]. Small modifications will then be made to contain the filters [2.3.4] within the holder. Also, a surface will be added below the camera holder

to reduce the bending moment. This was from the resulting cantilever beam that was made in the connection between the filter holder and servo shaft.

During the development of the filter holder system, the enclosure to hold the payload and autonomous system was being developed. A meeting with those at ACI assisted in outlining some requirements of the bus given the mission parameters for a flight test.

The bus itself was designed in SOLIDWORKS [7] to have the form factor of a 1.5U cubesat. Walls of the bus were made to be 3.5 mm thick. This provided enough support to the structure while giving space for all the components. One of the sides is removable and contains a shelf which rests on indents in the walls of the bus. The three main components needed inside the bus are the filter [2,3,4] and camera [8] payload system, the power source [10], and the Raspberry Pi computer [9]. Once the filter system was properly downsized, a tight arrangement of the payload with the camera [5] facing downwards was created. The battery [10] was placed on the top shelf of the bus due to its large size. The placement is shown in the SOLIDWORKS [7] assembly shown in Fig. 8.

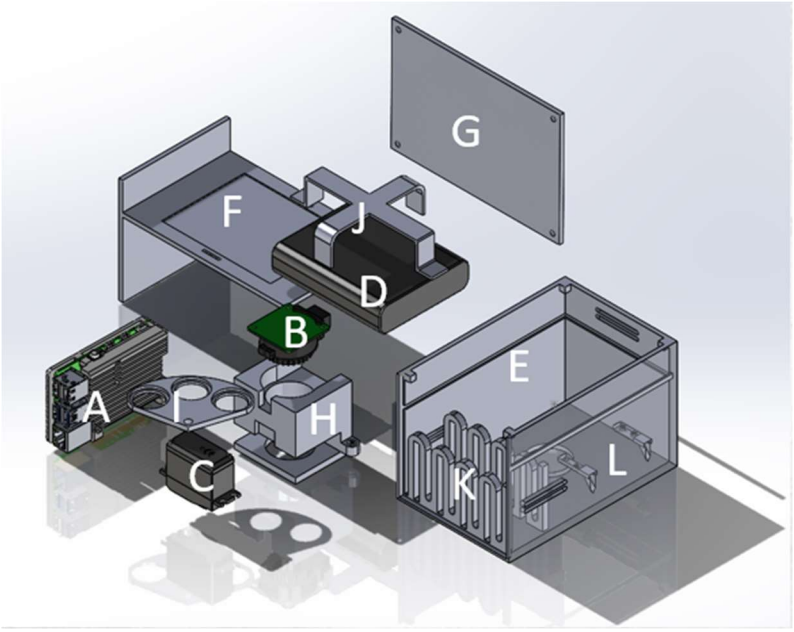


Fig. 8 SOLIDWORKS assembly of the payload and its enclosure. *A) Computer [9] with heatsink. B) Camera [5]. C) Servo [8]. D) Battery [10]. E) Payload bus. F) Removable wall with shelf. G) Removable cover for bus. H) Camera holder. I) Filter holder. J) Battery mount. K) Mount for computer. L) Mounts for servo.*

A communication device will eventually be attached onto the computer to make the system autonomous and send data back to the ground. There are holes in a few of the sides to provide ventilation and disperse the heat generated by the computer [9] and battery [10].

IV. Burn Test Results

A. Burn Test Data

The initial wood burning tests were focused on using 770 [3] and 780 [4] nm filters, 1.0 and 1.0 gains, 200 SS, and 200 ISO. Five pictures of each fire were taken per filter at five feet from the camera. This made for a total of ten images per fire and 110 images for the entire set. The wood was positioned against a non-reflective, blank wall and was lit only by natural light. These images were input into the saturation image processing script and observed to have no saturated pixels. In fact, the images are quite dim as seen in Fig. 9. This initial set of data used a variety of wood and served only to adjust the saturation of the camera to a real wood fire.

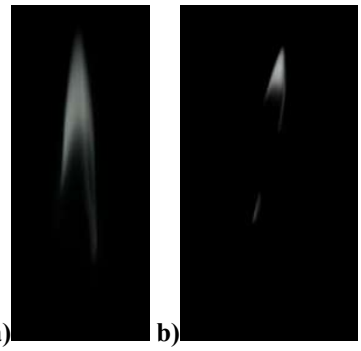


Fig. 9. Wooden dowel fire images using 1.0 and 1.0 gains, 200 SS, and 50 ISO.
a) Image using 770[3] nm filter. b) Image using 780[4] nm filter.

The current wood burning tests are dedicated to analyzing a layer, or slice, of each fire. A layer constitutes a row of pixels and their respective saturation values. The distance between the camera and the fire was recorded for each test. The types of wood were also monitored. The gains remained the same despite the SS and ISO changing to 500 ms and 200 ISO, respectively. This will provide a visual of the dip caused by oxygen and the subsequent spike of potassium light emissions. In other words, it will help identify potassium light from other burning elements. The slicing process used pictures taken with the 760 [2], 770 [3], and 780 [4] nm filters along with the same camera settings as the initial testing. Over 200 images were acquired and analyzed in this way, and a method of detecting potassium was devised. These images were of burning pine needles, packing paper, table salt, and potassium chloride. Even without the image processing, a trend in potassium can be seen. The table salt does not contain potassium, and therefore has almost no visible light under the filters [2,3,4], other than the heated metal tool used to hold the salt. However, pine needles, paper, and potassium chloride have a visible fire shape. Images using the 770 [3] nm wavelength filter of each burn test can be found in Appendix B.

B. Image Processing

In order to obtain an accurate measurement of meaningful data, the images taken with the camera [5] were analyzed using MathCAD [12]. The MathCAD [12] program included a series of mathematical functions that were capable of reading an image and producing several plots and histograms. As previously mentioned, the chosen method for detecting wildfires was by observing potassium spikes in the near-infrared images. To verify this method, large data sets were collected by taking hundreds of pictures of different burning materials.

The histograms generated represented a measure of the luminous intensity over wavelength. The luminous intensity was normalized to have a maximum of 1 for an improved visualization of the data. The peaks around the 760nm wavelength were caused by burning potassium. The peak in luminous intensity around the desired wavelength allowed for digitization of the flame images. With this process, the flame was able to be rebuilt and plotted in MathCAD [12] while removing all other components of the image taken. During the processing of these images, a potential problem was encountered. Since sunlight is emitted at such a large wavelength range, 400-750 nm, it was picked up by the camera [5] and outlined the brick background in the first plot in Fig. 10.

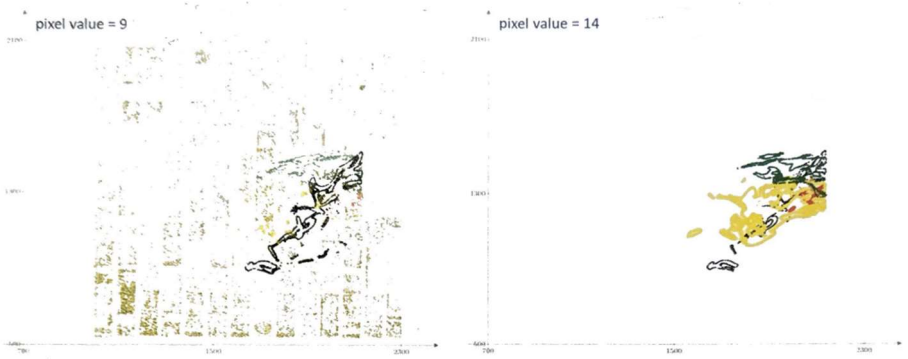


Fig. 10 Demonstration of sunlight effects using digitization of burning pine needles.

The sunlight was blocked out by filtering out the fainter pixel values emitted by the ultraviolet rays.

Recently, a more concrete method of averaging the image data, referred to as the 3-point method, was created in order to detect the burning potassium. Data taken from a wildfire using a hyperspectral instrument and lab experiment of imaging burning pine needles using a spectrometer from Ref. [1,13], shown on the right side of Fig. 11, serve as a reference; the x- and y-axis for these plots represent spectral radiance and wavelength, respectively.

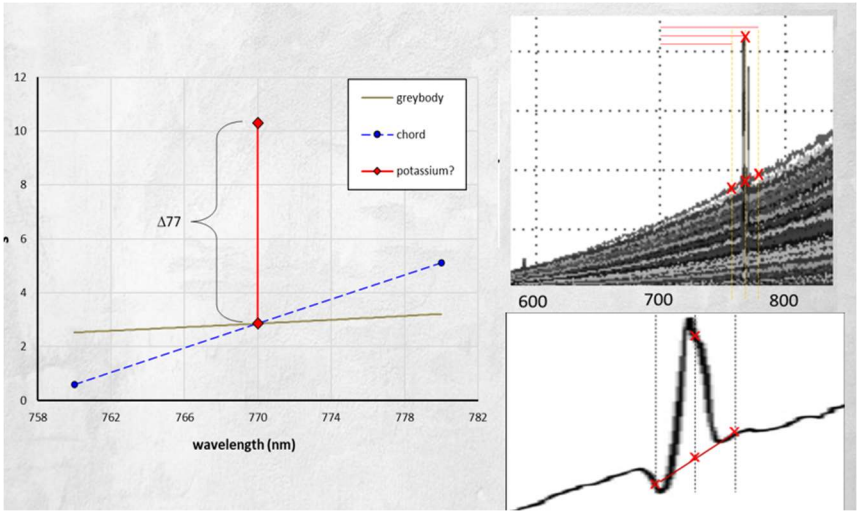


Fig. 11 Graphical representation of the 3-point method.

The dotted line, referred to as the chord, connects the three chosen points at 760 nm, 770 nm, and 780 nm. A range of pixel values from about 150 to 255 was used to filter out the black and less bright pixels. From there, arrays of sums of pixel values *s*76, *s*77, and *s*78 were created for the wavelengths at each point.

The midpoint of the chord, *s*77m, is established by averaging the sum of the arrays *s*76 and *s*77 for the desired index of pixel values. From there, the difference between *s*77 and *s*77m is take and represented by Δ77. The sign convention of Δ77 determines whether the burning substance contains potassium or not with a positive value indicating the burning of potassium. Figure 12 below shows the method applied to burning pine needles and packing paper using the camera and filter system.

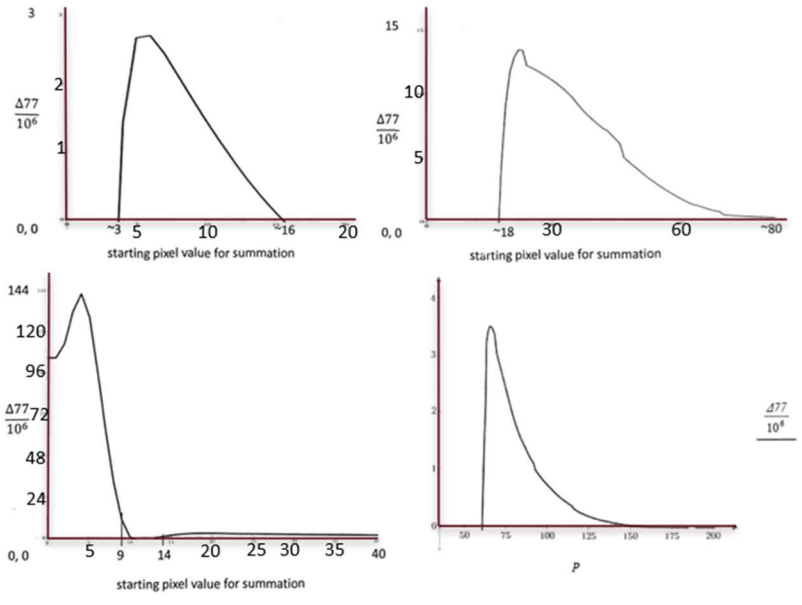


Fig. 12 Δ77 results from burning pine needles and packing paper using the 3-point method.

All show positive values meaning potassium is being burned. This method was then defended when an analysis of burning cardboard showed no positive Δ77.

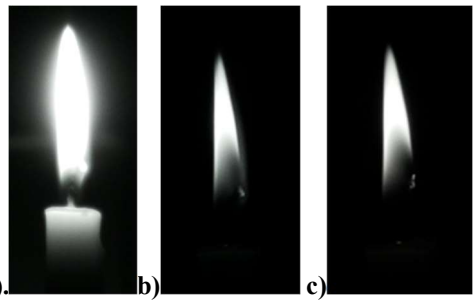
V. Conclusion

Throughout this project, experiments were conducted to collect data from burning vegetation. When beginning the data acquisition process, it was found that 1.0 and 1.0 gains limited the RGB values being assigned to each pixel. This was important to analyze captured images in the near infrared spectrum. Because of this, a Python [6] program was written to take a series of images. MATLAB [11] and MathCAD [12] were used to process these images. Experience using microcontrollers was gained by altering the code for the camera [5] and observing trends between the captures. From there, shutter speed and ISO were altered using Python [6] image settings which provided familiarity with camera settings. Once parameters in the code were finalized, burn testing with biomass, notably pine needles, began. This provided a consistent set of data for analysis. This included user friendly code for experiment repeatability, and the 3D printed ground testing models designed in SOLIDWORKS [7].

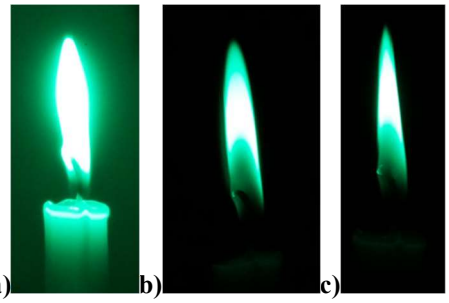
In preparation for UAV and cubesat integration, a compact mobile assembly was developed. This included the payload system utilizing the filter holder, camera [5], and servo [8]. The other system components were also fitted into the enclosure. Everything was placed into a SOLIDWORKS [7] assembly. The ground testing Python [6] program was adapted for autonomous functionality. This new system will advance from ground testing to in-air testing by UAV. Finally, the payload will progress to cubesat implementation in low earth orbit (LEO) and detect wildfires in the United States.

Appendices

A. Images From Candle Saturation and Gain Variation Tests

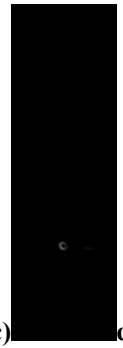


Candle using 1.0 and 1.0 gains, 2000 SS, and 200 ISO. *a) Candle without filter. b) Candle with 770 nm filter. c) Candle with 780 nm filter.*



Candle using 0.5 and 0.75 gains, 2000 SS, and 200 ISO. *a) Candle with no filters. b) Candle with 770 nm filter. c) Candle with 780 nm filter.*

B. Image From Wood Burning Tests

a)  b)  c)  d) 

Burn tests using 1.0 and 1.0 gains, 500 SS, and 200 ISO using 770 nm filter. *a) Pine needles. b) Printer paper. c) Table salt (NaCl). d) Potassium chloride.*

Acknowledgments

We thank our advisors Dr. Keith Koenig and Mr. Robert Wolz for guidance and support.
We also thank the Advanced Composites Institute (ACI) for providing their 3D printing capabilities.
Special thanks to Cody Hardin, Hunter Watts, and Luke Salisbury at ACI.

References

[1] Amici, S., Wooster, M.J., Piscini, A., "Multi-resolution spectral analysis of wildfire potassium emission signatures using laboratory, airborne and spaceborne remote sensing", Remote Sensing of Environment, Vol. 115, 2011, pp. 1811-1823.
[2] 760 nm filter, https://www.thorlabs.com/thorproduct.cfm?partnumber=FBH760-10
[3] 770 nm filter, https://www.thorlabs.com/thorproduct.cfm?partnumber=FBH770-10
[4] 780 nm filter, https://www.thorlabs.com/thorproduct.cfm?partnumber=FBH780-10
[5] Raspberry Pi High Quality Camera, https://www.raspberrypi.com/products/raspberry-pi-high-quality-camera/
[6] Python, PyCharm IDE, Software Package, version: PyCharm 2023.1, JerBrains, 2023
[7] SOLIDWORKS, Dassault Systemes, Software Package, SOLIDWORKS 2022, Hirschtick, Jon, 2022
[8] TowerPro SG-5010 servo, https://www.adafruit.com/product/155
[9] Raspberry Pi 4 Model B, https://www.raspberrypi.com/products/raspberry-pi-4-model-b/
[10] Anker PowerCore battery, https://www.anker.com/eu-en/products/a1215?variant=41510221283518
[11] MATLAB, The MathWorks Inc., Software Package, version: 9.13.0 (R2022b), Natick, Massachusetts 2022
[12] MathCAD, Parametric Technology Corporation, Software Package, MathCAD Prime 9, Massachusetts 2024
[13] Castillo, F., et al, "Study Spectral Emission of Burning Biomass in the VIS and NIR Spectral Band", Conference Paper, CHILECON, Dec. 2017.