Spectral Imaging System
CS294-164: Computational Color

Abinaya Dinesh
University of California, Berkeley
adinesh@berkeley.edu

Andrew Shen
University of California, Berkeley
shenandrew@berkeley.edu

Ashley Chu
University of California, Berkeley
ashchu@berkeley.edu

Jaewon Lee
University of California, Berkeley
jaewon_@berkeley.edu

***Abstract*—The Spectral Imaging System (SIS) project is an exploration diving into our curiosity of understanding the world through the information we can't see. Multispectral imaging integrates infrared (IR) and ultraviolet (UV) spectrums with traditional RGB imaging in a novel pentachromatic system. This study developed a multispectral camera system that captures a wider range of light, providing an understanding of our physical environment beyond human capabilities. The project employed advanced camera configurations and specialized filters for effective IR and UV capture, combined with sophisticated image processing and alignment techniques. Principal Component Analysis (PCA) and a Computational Neuroscience (CompNeuro) model were utilized for dataset analysis, highlighting the unique contributions of IR and UV data. These methods enhanced color recognition capabilities and offered novel insights into human color perception. Our re-colorization techniques blended scientific exploration with artistic expression, showcasing the potential of multispectral imaging in diverse applications, from environmental monitoring to artistic creation. The SIS project lays a foundation for future advancements in multispectral imaging, promising significant contributions to both science and art.**

1. Introduction

Multispectral imaging expands the capabilities of the human visual experience to reveal visual information invisible to the human eye. By harnessing the power of a wide spectrum of wavelengths, from near-IR to UV, our experiment aims to showcase the versatility of multispectral imaging in everyday settings, highlighting its potential to rethink about how we perceive and interact with our environment. From analyzing the health of urban flora to uncovering hidden details in buildings, the applications are as diverse as they are profound. However, multispectral cameras are often extremely costly and difficult to access. Currently, traditional approaches to multispectral imaging include using filters to allow specific spectrums to pass through a sensor, using a beam splitter to split wavelengths of light to different sensors, or having band filters integrated into a camera sensor's configuration. Still, the amount of multispectral data of everyday space is sparse, with most datasets featuring aerial photography or only small subsets of data featuring day-to-day life. We sought to find a solution through the use of accessible multispectral imaging supplies and implementing techniques to enhance our understanding of the daily world around us through capturing, validating, and reimagining a multispectral dataset. By applying image processing techniques and establishing a public image processing pipeline for our dataset, we hope to take new steps towards capturing, sharing, and exploring the world of color with the world.

2. Related Works

The first handheld camera that could capture information in the IR range of the electromagnetic spectrum was invented in 1910 by an American physicist named Robert Williams Wood, whose photography styles required experimental film and long exposure times. Yet, its incredible uses during World War 1 and love from curious photographers led it to become rapidly commercialized by companies like Kodak, particularly with the invention of their Ektachrome Professional Infrared EIR film (the military version being Aerochrome III).

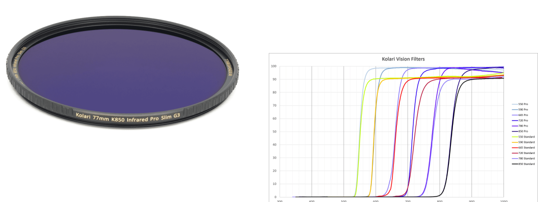
After the Ektachrome film stopped being manufactured, the photography community turned to online editors and recolorization techniques to incorporate the black and white channels of IR into their RGB channels, but experienced a plethora of issues related to color fringing during image capture and the limitations of custom, beta color mixing technologies (such as the unsupported IRG Image Transform V1.2). While spectral imaging has become popularized over the years with forms of IR photography both on film and digital mediums, spectral imaging has often been very timely and expensive with the equipment and devices needed. The large majority of commercial IR photography cameras range between $4,000-$10,000 according to a brief search on Google Marketplace. There are several projects that have used alternative lower cost imaging supplies such as Raspberry Pi camera modules to capture these tools. Past studies at the University of Barcelona have used RGB cameras to create multispectral data via using the camera to capture fruit in different wavelengths of light [10], for aerial photography [9], and for hobbyists to create personal IR camera [18]. However, we were unable to find many photography groups or techniques actively integrating both UV and IR data together.

3. Multispectral Camera Experimentation

Figure 1: Monochrome and RGB StereoPi setup

Cameras allow us to capture the world in 3D colorspace. To do this, cameras capture light in the visual spectrum, from 400nm to 700nm, by using a color filter array placed on top of an image sensor. With the assistance of our advisor, Atsu Kotani, we constructed a method for capturing wavelengths outside of the visual spectrum.

3.1. IR Capturing Technique



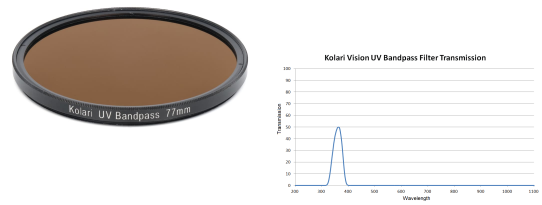
(a) Kolari Standard 850nm filter lens

(b) Transition Curves

Figure 2: IR Filter Lens

On top of the color filter array, an IR blocking filter is usually placed on top as a filter to block IR light. The addition of an IR blocking filter ensures that an image is truly accurate according to the human visual system. With Raspberry Pi's camera modules offering both standard representation and a no block IR filter, we created a new multispectral camera system. We began our construction with a StereoPi v2 board, a Raspberry Pi v2 camera, and a Raspberry Pi v2 NoIR (without IR blocking filter) camera. Using the novel technology of the StereoPi v2 board, StereoPi provides a dual camera GUI capture software to add customizations and control parameters such as brightness and exposure. Both cameras have a 3mm focal length and aperture of f2.0, allowing for direct alignment and processing. Once constructed, we can apply an 850m standard Kolari IR long-pass filter to the IR camera to capture two images in real-time, one RGB and one IR.

3.2. UV Capturing Technique



(a) Kolari UV filter lens

(b) UV Filter Band-pass Filter Transmission

Figure 3: UV Filter Lens

After building our initial RGB-IR camera, we wanted to explore the idea of capturing 5 channels of information, leading us to add an additional UV camera to capture UV data. However, for building a UV camera, new difficulties arose. While color filter arrays are necessary to reconstruct color from a single capture, they also absorb a large amount of UV light. The removal of the color filter array is extremely difficult with the main techniques consisting of laser removal, chemical removal, or physical removal, often done by scratching the filter off. Luckily, MaxMax provided a deep Raspberry Pi HQ UV camera without a color filter array or IR blocking filter, allowing us to capture UV light. By adding a new camera, we needed to redesign our imaging system by capturing our visual and IR data with HQ cameras to maintain similar focal lengths and apertures. Both cameras used lenses with a 6mm focal length and aperture of f2.8.

3.3. Camera Limitations and Problem-Solving Techniques

There were two main challenges to our approach of using several cameras. First, we must capture the images instantaneously. Second, the configuration settings for both cameras defaulted to one camera as arranged by the StereoPi system.

To overcome the challenge of capturing images instantaneously with multiple cameras, we developed a synchronized shooting system and added a physical button to toggle captures. This setup was crucial to ensure that the images from the different cameras (RGB, IR, and UV) were captured at the same moment, thereby maintaining consistency across the different spectral images. The synchronization was achieved through custom software scripts that controlled the camera triggers and by wiring a button component for simultaneous activation.

For the second difficulty, we modified the configuration settings to manage each camera independently within the StereoPi system. This modification allowed us to adjust settings such as exposure, gain, and white balance for each camera separately, which was essential due to the different light sensitivities and spectral characteristics of the IR and UV cameras compared to the standard RGB camera. By fine-tuning these parameters, we were able to achieve a uniform exposure across the different spectral images. Additionally, we integrated the use of a neutral density (ND) filter to achieve automatic lighting adjustments.

The integration of the RGB, IR, and UV cameras into a single system provided us with a robust multispectral imaging platform. This setup enabled us to capture a more comprehensive range of spectral data, significantly enhancing our ability to analyze and interpret the world around us in new ways.

4. Capturing the Dataset

In this SIS project, our objective was to develop a comprehensive multispectral dataset. We focused on capturing static outdoor scenes in the urban and natural environments of UC Berkeley and Alameda. This approach diverges from traditional satellite hyperspectral imagery and provides a unique, ground-level spectral perspective.

4.1. Camera Setups and Spectral Data Acquisition

We employed two tailored camera setups for specific spectral data acquisition. The initial setup comprised StereoPy and two wide Raspberry Pi Cameras (PiCams), capable of capturing RGB and IR images. A significant challenge was balancing the exposure values between RGB and IR images. To address this, we implemented a Neutral Density (ND) filter on the RGB camera, enhancing exposure balance.

Subsequently, we introduced an advanced setup featuring StereoPy and two High-Quality (HQ) Cameras. This configuration expanded our capabilities to include UV imaging and provided enhanced control over focus and exposure settings. The monochrome HQ camera, notably versatile, facilitated seamless switching between RGB+IR and RGB+UV imaging modes.

4.2. IR and UV Filter Selection and Efficacy

IR Imaging: We selected the Kolari K850 filter for its optimal transmission properties at 850nm, ideal for effective IR light capture. This filter was crucial in our IR imaging setup, blocking visible light to ensure true IR spectrum representation in the captured images.

UV Imaging: The Kolari UV Bandpass filter was our choice for UV imaging. Known for its peak transmission at 365nm, this filter efficiently captures UV light, maintaining over 25% transmission between 345-380nm. Its high total light transmission capability was instrumental in our UV imaging process.

4.3. Data Capturing Methodology and Conditions

Our data capturing focused on stationary objects, aligning with the static nature of our setup. We ensured optimal lighting by conducting photography sessions at solar noon, when the sun was at its zenith. To maintain stability and consistency in image capture, a tripod was used across all datasets.

4.4. Multispectral Dataset

Over the course of our data collection, we captured a variety of outdoor images from Berkeley and Alameda of buildings and natural life. As a result, we have produced three datasets with aligned images that are available for download on our website where they can also be visualized in three dimensions using our in-browser editor, which will be discussed in the Real-Time Recolorization Editor section. A breakdown of the various datasets is provided in Figure 4.

Each multispectral image is stored in NumPy array that has dimensions $width \times height \times \#$ of channels, where the channels in the order of blue, green, red, IR, and UV. If an image has only 4 channels, then all channels except UV will be included. The RGB, IR, and UV channels of a multispectral photo is shown in Figure 5

5. Alignment

Due to the nature of the cameras set up on SIS, there are slight physical position differences between the cameras which will give each camera a slightly different viewing angle. As a result when trying to create a multispectral image, it does not suffice to simply stack the image as the misalignment is far too large to be ignored. To solve this problem, we begin by extracting features from each of the visible, IR, and UV images. Initially, we began by using the open source ORB [14] feature detector to find features, however despite some initial successes, we determined that in many cases it did not successfully find features in the IR and UV images. This is as a result of an combination of few challenges, primarily being that the IR and UV images were either too dark due to difficulties in properly adjusting the exposure as previously mentioned in the camera and configuration section or due to the color differences as the IR and UV images are cast to grayscale. Ultimately, we used the SuperGlue [15] pretrained network which is another feature detector utilizing graph neural networks.

After computing these feature pairs, we used RANSAC [**?**] homography to select good feature pairs to compute a homography matrix to warp the IR and UV images into the same viewing angle as the RGB image. Lastly, we combine

Datasets			
Dataset Name	# of Channels	Image Dimensions	# of Images
Berkeley RGB_IR_UV.zip	5 (R, G, B, IR, UV)	2028×1520	22
Alameda RGB_IR.zip	4 (R, G, B, IR)	1296x972	16
Berkeley RGB_IR.zip	4 (R, G, B, IR)	1640x1232	53

Figure 4: A description of each dataset

(a) RGB

(b) IR

(c) UV

Figure 5: RGB IR UV Images of the Campanile

all of the images which are now all aligned by stacking them into a three dimensional NumPy array where the dimensions are width by height by channels. One caveat of using the homography method is that when we warp the IR and UV images, there are some regions that do not overlap with the visible image resulting in some regions of the multispectral image having no data in either the IR or UV channels. This makes sense because because each camera is pointed in a slightly different direction, there are some regions that

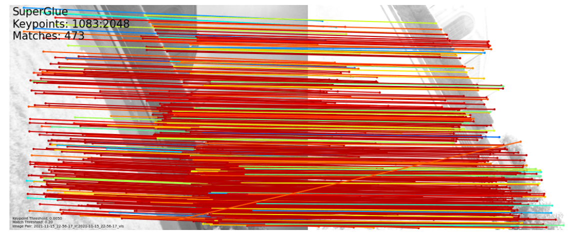


Figure 6: Side by side RGB and IR images with colored lines connecting feature pairs generated by the SuperGlue pretrained network

are only captured by one camera. We used the Python `jupyter_compare_view` package to help visualize the multispectral area with a convenient display that shows IR regions on mouseover.

6. Recolorization

In our SIS project, we have broken through the limits of conventional imaging. Our recolorization process was not just a technical endeavor; it was a journey to broaden the spectrum of human perception. Inspired by the color vision of human tetrachromats and animals like hummingbirds, and pigeons, we provide a sight into what observation of our natural world would be like with access to information beyond the visual spectrum in a beautiful, eye-catching way.

6.1. The Art and Science of Recolorization Techniques

Weight and Bias Adjustment: This technique uses an invisible spectrum of light as a variable. We adjusted the RGB channels by incorporating IR or UV data, creating a formula for transformation:

$$R' = norm(IR) \times R + norm(IR) \times c \tag{1}$$

and similarly for G' and B'. This approach allowed us to enhance the image quality with nuances.

Color Mapping Technique: In this phase, our artistic endeavor took center stage. We first split each RGB, IR, and UV image into separate monochrome layers. These layers were converted into halftone images with a resolution of 1500 dpi, enhancing their textural detail. Subsequently, we assigned specific colors to each halftone layer, carefully overlapping them. This process transformed the unseen IR and UV spectrums into a visually vibrant collage, effectively merging the obscure beauty of these spectrums with the tangible realm of visual artistry.

RGB Layer Modification: We replaced the traditional RGB framework. By substituting channels like green with IR data, we illuminated aspects of scenes usually concealed from the naked eye. This technique highlighted elements responsive to IR or UV light, such as specific surface textures.

PCA Recolorization: Another method that we used to approach the channel of having more than 4 channels available but only 3 visible channels, is to use PCA by treating each pixel as a 4 dimensional point in space. Then using PCA, we can take the first 3 principal components and flatten all of our multidimensional points down to 3 dimensions so that they can be displayed in the RGB color space.

(a) Weight and bias adjustment

(b) Color mapping technique

(c) RGB layer modification

(d) PCA recolorization

Figure 7: Photograph of a tree with different recolorization approaches

6.2. The Fusion of Science and Art

Our recolorization techniques included a harmonious blend of scientific discovery and artistic expression. By translating the IR and UV light into visible rays, we offered new perspectives in Figure 7. These methods provided a novel lens through which to view the natural world, marrying scientific utility with the splendor of visual art.

7. Real-Time Recolorization Editor

In order to display our results in an accessible and understandable manner, we built a full stack website using React and Typescript for the frontend and Flask for the backend. The web application hosts a gallery of our dataset images with the added functionality of being able to click on an image which brings the user to an editor as pictured in Figure 8.

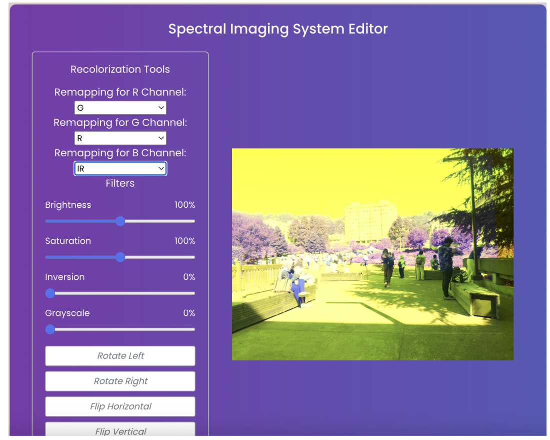


Figure 8: Real-Time Recolorization Editor

The editor has a number of basic tools that allow the user to adjust the brightness, saturation, grayscale, and rotate or flip the image. Moreover, the editor provides features to reassign the red, green, and blue channels for the images, allowing the user to colorize the image using the IR and UV channel information. We aimed to optimize the experience for creating artistic interpretations and reimaginations of daily life in the Bay Area. This source code and images for the website can be accessed at `https://github.com/andrewtshen/sis-website` and also allows for easy access and download to the datasets we have collected in the project. The website is still a work in progress as we continue to add more features and ways to recolorize images.

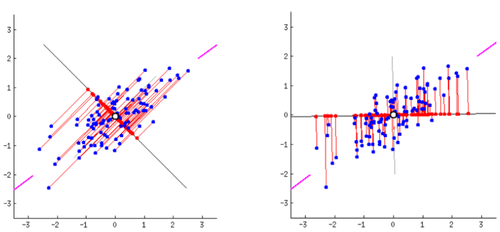
8. Evaluating the dimensionality of our tetrachromatic dataset

By design of our stereoscopic setup, the two cameras should collect linearly independent information of the physical world. The camera from Figure 1 collects light strictly in the 300-700nm range as a result of a IR cut filter built into the Raspberry Pi HQ, while the camera on the right, which has the IR cut filter removed and an additional band-pass filter, has transmission only above 850nm, enabling us to collect light in the IR spectrum. While these transmission curves help us understand the physical conditions of our setup, it is important to evaluate the color dimensionality of resulting images taken by the camera.

In the following sections, we describe two methods for evaluating dataset dimensionality, the first method being Principal Component Analysis (PCA) and the second being a computational neuroscience (CompNeuro) model [?] that is able to use a sample 4 cone retinal cone mosaic to learn spectral sensitivity curves for each of the cones. Evaluation of the CompNeuro model relies on classical color matching tests to evaluate how much of a target color (in our case, a patch of the image) can be accurately described by independent light sources (the individual cones learned by the model). We are able to show via PCA that the RGB+IR taken by the SIS truly requires an additional principal component to represent when compared to an RGB-only counterpart, while the RGB+IR+UV image requires 2 additional principal components. We are also able to show preliminary results from training the CompNeuro model which show that three cones is only able to represent 60% of the information of the original dataset, and an extra cone is needed to view more than 90% of the information represented in the images taken by this camera.

8.1. Statistical Analysis of Dataset Dimensionality

PCA is an algebraic method that attempts to represent as much of the information in an input dataset in as few components as possible. It does this by looking at the relationships between all pairwise features in the data (for us, it is the individual channels of the image: RGB and IR) and finding a linear combination of the features such that, after projecting all original points onto the line, the variance of points along the line is maximal and the difference between the original point's position and its new position is minimal. In two dimensions, this can be thought of as finding a line of best fit between two variables for the first principal component, and as the dimensions increase, the linear combination may be of higher degree than 2. Each principal component is constructed by finding the line which maximizes the variance of points when they are projected onto the component, as exemplified by Figure 9.



(a) A visual explanation of the process of finding the first principal component for a 2D spread of data points. Red bars indicate the distance of the point from its projection on the line. Purple lines are the optimal line which minimizes error between original and projected points while maximizing the variance of the points along the line (spread of points after projection), so as to prevent information lost

(b) The process of finding the second principal component, now maximizing for variance while minimizing distance between original and projected point along the line orthogonal to the first principal component.

Figure 9: PCA

8.2. Results from Principal Component Analysis

These results are still earyl results as our PCA method is still not fully evaluated because we have only tested a small subset of dataset images. However, this is an area we are actively working on. We use PCA to identify the minimum amount of components we need to describe 99% of our original dataset. For a single image, the pipeline of PCA can be broken up into 4 steps. First, we mean center the image to normalize the data values between 0. Then, we create a covariance matrix of the image by finding the pairwise covariance between each channel against each other. The shape of the resulting matrix for a RGB+IR image is 4x4, where the diagonal of the matrix is simply Cov(X, X), or Var(X), where X is a channel of the image. Next, we calculate the eigenvectors and eigenvalues for the covariance matrix. The eigenvalues can be thought to represent how much of the original image is represented by its associated eigenvector, which is used in the next step of calculating how many eigenvectors are needed to reconstruct the original image with 99% accuracy. Finally, we divide the amount of information described by each eigenvector by the total variance described by all the principal components.

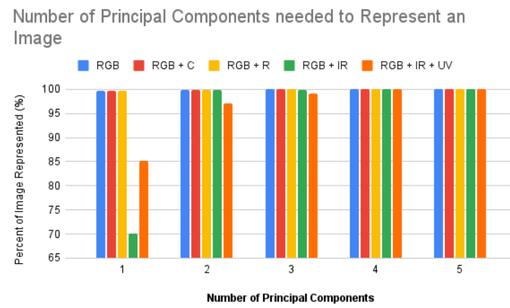


Figure 10: The principal components needed to represent each dataset

In order to evaluate the results of PCA on our data, we set up baseline experiments with images composed of RGB channels, RGB+R (where the red channel is duplicated as a 4th channel), RGB+C (where C is a scalar value for the fourth channel), RGB+IR (the results from our tetrachromatic camera setup) and RGB+IR+UV (the results from our pentachromatic camera setup). Results from this 10 show that the RGB, RGBR, and RGBC images can all be represented by a single principal component, while the RGB+IR image requires two principal components, and the RGB+UV image requires three principal components to reconstruct 99% of the original data. This is sufficient for us to conclude that the UV and IR camera setups have actually captured new information not possible by a normal camera capturing light in the visible spectrum. However, more spectral analysis of the resulting images and channels would be required to see how much of each wavelength of light in the UV and IR channels are actually being represented in the resulting images.

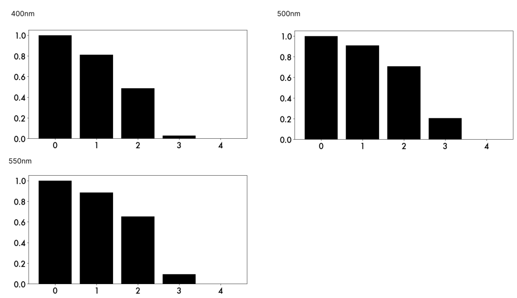


Figure 11: Degeneracy testing results

8.3. Biological Model of Color Dimensionality Using Computational Neuroscience Model

Past work [?] in developing computational models of color vision from visual sensorimotor data enables us to leverage a self supervised model that can learn a retinal cone mosaic capable of viewing any input image in full color dimensionality. This model works by taking in normal camera images (with arbitrary channels), projecting these against a given retinal cone mosaic to obtain optic nerve signals, then learning cone identity functions for up to 5 cones (dependent on specifications) and a demosaicing network that functions as a predicted retinal cone mosaic for the "viewer" of the input dataset. The output of this model is a high dimensional internal percept which we can view using Neural Scope to compare the reconstruction of the original image against the ground truth input image, where the model has only learned from the optic nerve signals obtained by the image.

8.4. Modeling on RGB IR dataset

To train our model on images captured from our RGB+IR camera setup, the model takes our directory of 104 four channel images (RGB+IR) of dimension 1232x1640 and converts this into 50,000 randomly sampled patches of 37x37 images to become the final training dataset. Only 69 of these images are displayed in the final web page and made publicly available, so as to protect the privacy of students captured in the photos. To evaluate the results of the model, we run a classical color matching test, where the model calculates the mean squared error between the internal percepts of two colors, one being a target color and the other obtained by modifying the weights of n primary light sources. In our case, to evaluate the number of cones the model needs to recreate a color from the input image, we try running color matching functions with 1 through d primary light sources and calculate the error for each, where d is the predicted dimension of the input image. When the error between internal percept and target color is $\leq 10\%$, we observe the number of primary colors as being the number of different cone types necessary to view the original image in full color dimensionality.

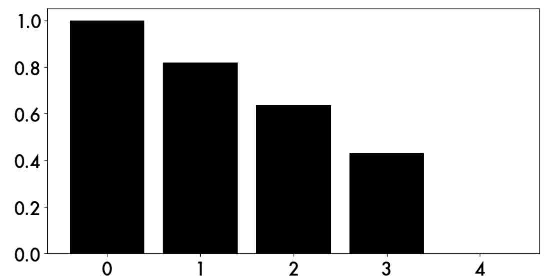


Figure 12: Results from tetrachromatic modelling of RGB + IR dataset

Results from conducting color matching experiments on the trained model of RGB+IR input images show that there could exist an organism which is able to view the image in full color dimensionality using 4 different cone types. Figure 12 shows these the 4th cone (0 index) as having greater than 10% error.

8.5. Degeneracy Testing

We conducted degeneracy testing on the computational neuroscience model to ensure that the results of the model were truly because of the additional range of spectra in the data, instead of a simple relic of a four channel NumPy array or tensor. To do this, we took our dataset of 104 RGB + IR images, but replaced the 4th IR channel with a copy of the R channel, making these RGB+R images. This image only has 3 unique channels of information, and after testing the model on these images at Q peaks at 400nm, 500nm, and 550nm, the model consistently trains to trichromacy, indicating the results from the tetrachromatic input were truly as a result of new information being provided. Figure 11 shows results from the degeneracy testings.

9. Conclusion & Future Work

The Spectral Imaging System (SIS) project aims to help progress the field of multispectral imaging. At its core, the project is about expanding the capabilities of conventional imaging systems by integrating not just the visible light spectrum (RGB) but also infrared (IR) and ultraviolet (UV) spectrums.

On the hardware and capturing side, though we were able to create a strong foundation for our camera, we are still at the cusp of capturing data that is fully clean, organized, and configured automatically to take large amounts of photos quickly and in a portable manner. With the Raspberry Pi HQ camera, the camera comes with a rolling shutter that can be difficult to use, as this causes a slower shutter speed. Additionally, we are still manually switching our Kolar near IR and UV filters manually. By installing a second Raspberry Pi HQ camera without an IR blocking filter, we hope to capture this data more accurately and instantaneously.

Due to the iterative developments of the camera during our research process, our dataset has also been inconsistent, featuring a range of buildings, outdoor scenes, and foliage. As our camera is now more established, we hope to have more uniformity in our images on calibration where sizing, lighting, and datatypes are all consistent.

A crucial component of our project was the advanced image processing and alignment methodologies. We utilized sophisticated algorithms, including feature extraction and homography techniques, to address the inherent challenges in aligning multispectral images. These images, captured from different spectral cameras, often exhibited slight variations in perspective due to their physical setup. Our approach involved extracting key features from each image using advanced algorithms like the SuperGlue pre-trained networks, which utilizes graph neural networks for feature detection. This allowed us to accurately align images from different spectrums, ensuring that each pixel corresponded to the same physical point in space across all spectrums. Moreover, the application of neural network-based techniques facilitated the refinement of these alignments, enhancing the overall quality and accuracy of the multispectral images we produced. This meticulous process was instrumental in maintaining the integrity and reliability of the visual data, crucial for any subsequent analysis or application. Moreover, the project extensively utilized Principal Component Analysis (PCA) and a Computational Neuroscience (CompNeuro) model to analyze our rich dataset. The PCA method offered exciting insights and by being instrumental in reducing the dimensionality of our data, allowing us to identify the most significant components that capture the essence of the information across different spectra. By computing the eigenvectors and eigenvalues of the dataset's covariance matrix, we were able to discern how each spectral component, including IR and UV, contributed to the overall variance of the data. This analysis provided insights into the unique aspects of IR and UV data, which are not captured in traditional RGB imaging.

The CompNeuro model added another layer of analysis, offering a biological perspective on how multispectral data might be processed by the visual system. By simulating a retinal cone mosaic and learning cone identity functions, the model allowed us to understand how different spectral channels would interact in a biological visual system. This model was particularly effective in demonstrating how additional spectral information, like IR and UV, could enhance color perception and recognition beyond normal human capabilities. The use of these advanced computational methods significantly deepened our understanding of the multidimensionality and complexity of the captured images. In the realm of artistic expression and scientific utility, the recolorization techniques handled under the SIS project have shown new insight for visualizing and interpreting spectral data. The merging of scientific discovery with artistic creativity enhances the aesthetic appeal of the images.

In summary, the SIS project marks a considerable advancement in multispectral imaging, offering a comprehensive platform for capturing, processing and visualizing spectral data. Our methods have extended the boundaries of conventional imaging, providing enriched visual data that deepens our understanding of various environments.

10. Contributions

Ashley worked largely on the camera construction, web platform, camera scripting, and reviewing literature around IR imaging and recolorization techniques. She led the construction and exploration of the Raspberry Pi camera components and worked with Jaewon to capture datasets in Alameda and Berkeley. On the web platform, Ashley helped create the web interface with Typescript and React.

Jaewon was our certified photographer. He worked on setting up camera configurations, capturing data around Berkeley, and exploring weighted recolonization techniques. He took a main role in recolonization, balancing science and art. Ashley and Jaewon often collaborated together to explore new camera mounts, lenses, and camera configurations.

Abinaya spent her time heavily working on setting up our computational neural model to evaluate a tetrachromatic dataset and understanding how to validate it's dimensionality. She also worked on researching prior art in recolorization techniques, PCA analysis, and developing scripts for recoloring RGB data using the IR channel as weights with Jaewon.

Andrew led a large majority of our software development work. He set up our initial alignment and recolorization pipeline using SuperGlue and PCA. He also worked in collaboration with Abinaya to debug and initiate our computational neural model validation. Andrew also worked with Ashley to set up the React/Typescript website, and also scripted the backend to handle real-time recolorization in Flask.

Acknowledgments

This project was developed as a part of CS294-164 Computational Color at UC Berkeley under the support of Professor Ren Ng and his graduate students. We would like to thank Ren for his continuous support and mentorship throughout the journey of developing SIS. We would also like to thank Atsu Kotani for his unwavering support, emails at late hours, and encouragement through exploring the image capturing process and computational neural model analysis. Lastly, we would like to thank our fellow students who provided us with valuable feedback and advice throughout the semester.

References

[1] *Deep Ultraviolet Imaging Using the Raspberry Pi HQ Camera*, [Online]. Available: https://stereopi.com/blog/deep-ultraviolet-imaging-using-raspberry-pi-hq-camera.

[2] F. Durand, J. Dorsey, *Fast Bilateral Filtering for the Display of High-Dynamic-Range Images*, July 1, 2002, [Online]. Available: https://people.csail.mit.edu/fredo/PUBLI/Siggraph2002/DurandBilateral.pdf.

[3] *Feature Matching + Homography to find Objects*, 2011, [Online] Available: https://docs.opencv.org/3.4/d1/de0/tutorial_py_feature_homography.html.

[4] M. Fischler et al.*Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, 2023, [Online] Available: https://docs.opencv.org/3.4/d1/de0/tutorial_py_feature_homography.html.

[5] *Get Started in Infrared (IR) Photography*, [Online]. Available: https://www.adobe.com/creativecloud/photography/discover/infrared-photography.html?fbclid=IwAR1zNWvvbXvvqYhxXtFZYxC3PBs4qsQFaw0NhjzgRQ1X9KoabIQHiLyZsKM.

[6] *How Are Webb's Full-Color Images Made*, Feb. 2023, [Online]. Available: https://webbtelescope.org/contents/articles/how-are-webbs-full-color-images-made.

[7] Atsu. Kotani, *Unsupervised Learning of Color Vision Purely from Optic Nerve Signals*, Unpublished manuscript.

[8] H. MAÎTRE, *Statistical Properties of Images*, Nov. 2008, [Online]. Available: https://www.iste.co.uk/data/doc_noyoaiyjtdiw.pdf.

[9] D. Papp, Re-Creating the E. Belcore, *Raspberry Pi 3 Multispectral Low-Cost Sensor for UAV based remote sensing*, 2019, [Online]. Available: https://isprs-archives.copernicus.org/articles/XLII-2-W13/207/2019/isprs-archives-XLII-2-W13-207-2019.pdf.

[10] X. Allepuz Rico, *A Low-Cost Multispectral Camera*, Jan. 2020, [Online]. Available: https://diposit.ub.edu/dspace/bitstream/2445/176389/1/ALLEPUZ%20RICO%20XAVIER_474077.pdf.

[11] R. Pandey et al., *Learning to Relight Portraits for Background Replacement*, August 2018, [Online]. Available: https://augmentedperception.github.io/total_relighting/.

[12] Y. Philippe, *The IRChrome Infrared Photography Filter – Our Digital Version of Kodak Aerochrome*, January 29, 2019, [Online]. Available: https://kolarivision.com/the-irchrome-infrared-photography-filter-our-digital-version-of-kodak-aerochrome/.

[13] D. Papp, *Re-Creating the Unique Look of Unobtainable Aerochrome Film*, August 1, 2022, [Online]. Available: https://hackaday.com/2022/08/01/re-creating-the-unique-look-of-unobtainable-aerochrome-film/.

[14] E. Rublee, *ORB: An efficient alternative to SIFT or SURF*, 2011, [Online] Available: https://ieeexplore.ieee.org/document/6126544.

[15] P. Sarlin et al., *SuperGlue: Learning Feature Matching with Graph Neural Networks*, 2020, [Online]. Available: https://arxiv.org/abs/1911.11763.

[16] C. LeGendre et al., *Practical Multispectral Lighting Reproduction*, [Online]. Available: https://vgl.ict.usc.edu/Research/MultiSpectral/.

[17] *About the Camera Modules*, [Online]. Available: https://www.raspberrypi.com/documentation/accessories/camera.html.

[18] *A Raspberry Pi Multispectral Camera*, [Online]. Available: https://www.instructables.com/A-Raspberry-Pi-Multispectral-Camera/.

[19] *Snapshot Multispectral Cameras*, [Online]. Available: https://spectraldevices.com/collections/snapshot-multispectral-cameras.

[20] J. Zakaria, *A Step-by-Step Explanation of Principal Component Analysis (PCA)*, March 29, 2023, [Online]. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.