UNLV

HOWARD R. HUGHES
COLLEGE OF
ENGINEERING

EGG 499 - Research Internship for Embedded Engineering and Computer Science

By Sujin Park

Under the guidance of

Dr Venkatesan Muthukumar

Optimization of Lighting Conditions for Minimizing Shadows in Water Droplet

Abstract

This study aims to find optimal lighting conditions to minimize shadows during water droplet penetration experiments. Using the ESP32-E microcontroller for lighting control, the research proposes a methodology for shadow detection and analysis in images captured using the Raspberry Pi 4's 5-megapixel camera with subsequent processing using the OpenCV computer vision library.

1. Introduction:

This research focuses on minimizing shadows during water droplet penetration experiments.. The hardware configuration utilized the S500 Multi-Rotor Air Frame. ESP32-E, Raspberry Pi, and Raspberry Pi Camera were mounted on the top frame, while four NEOPIXEL LEDs were attached to the drone wing frame. The VEML7700 sensor on the ground measured light intensity to the plate using the LUX value. Illumination was maintained by controlling LEDs with PWM based on the measured LUX value. The Raspberry Pi 5-megapixel camera captured images of the plate for subsequent analysis. OpenCV was employed to detect shadows in the saved images and analyze their positions. Based on OpenCV results, LEDs connected to ESP32 were controlled to minimize shadows.

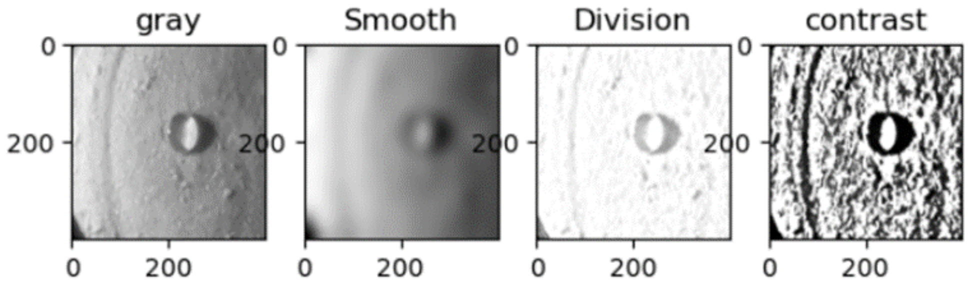
2. Methods:

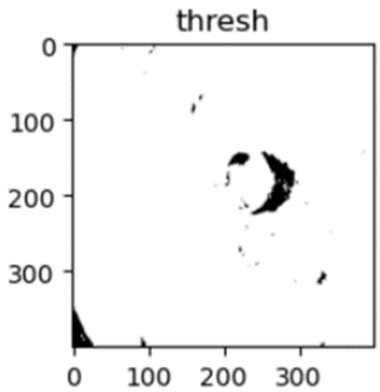
2.1 Illumination Measurement and Dynamic Control

Utilizing the VEML7700 sensor, the system measures the illuminance of light reaching the water droplet plate. Subsequently, the system dynamically adjusts and maintains a specific light level around the water droplet plate based on the measured LUX value. This ensures a consistent and controlled lighting environment, enabling precise observations and analyses during water droplet experiments.

2.2 Image Data Acquisition and Analysis:

In the process of acquiring and analyzing image data, the Raspberry Pi camera module is employed to capture and store images. Utilizing OpenCV, the captured images are converted to grayscale, followed by the application of a blur. The grayscale image is then divided by the result of the blur to enhance contrast, and subsequently, a binary thresholding operation is performed. Following this, the system calculates the number of dark pixels in different regions of the image, identifying the position where the darkest pixels are most prevalent. Through this image processing and analysis procedure, the system accurately identifies the intensity and position of shadows in the captured images.





2.3 Lighting Control System

The Adafruit NEOPIXEL LED comprises four strips, each corresponding to distinct regions. Leveraging OpenCV results, the system dynamically adjusts the brightness of the LEDs to effectively minimize shadows. Brightness control is achieved through Pulse Width Modulation (PWM), altering the brightness of the NEOPIXEL strips. This allows the system to intricately modulate the lighting, efficiently reducing shadows and enhancing the precision of experimental outcomes.

3. Hardware

1. Raspberry Pi 4 Camera

- Raspberry Pi 4 Camera Video Module 5 Megapixels

2. ESP32-E

ESP32-NEOPIXEL Connection

ESP32	NEOPIXEL
GND	GND
VCC	5V
D9, D12, D13, D7	Din

ESP32-VEML7700 Connection:

ESP32	VEML
GND	GND
VCC	3.3V
GPIO 21/SDA	SCL
GPIO 22/SCL	SDL

4. Conclusion:

The research showcases successful implementation of a lighting optimization system for water droplet penetration tests. By utilizing a combination of ESP32 and Raspberry Pi, the system effectively minimizes shadows, ensuring accurate measurement of WDPT and contact angles.

5. Code Implementation:

ESP32-E code:
https://github.com/soojin-p/UNLV_Project/tree/main/ESP32_Shadow_Remove

Raspberry Pi4 Camera code:
https://github.com/soojin-p/UNLV_Project/blob/main/RPi%20Camera.md

OpenCV code:
https://github.com/soojin-p/UNLV_Project/tree/main/OpenCV_Shadow_detect

6. References

Raspberry Pi4 Camera: https://www.waveshare.com/wiki/RPi_Camera_%28I%29#libcamera-still

ESP32-E: https://wiki.dfrobot.com/FireBeetle_Board_ESP32_E_SKU_DFR0654

OpenCV code: https://github.com/YalimD/image_shadow_remover