

Live-Feed-over-LAN Camera Spectrometer (LoLAN-CaS) Documentation

Members:

Andrea Rica Advincula

Creo Baylon

Kenneth Domingo

Rene Principe Jr.

Lou Josef Tan

1 Overview

The Live-Feed-over-LAN Camera Spectrometer (LoLAN-CaS) is an implementation of a spectrometer which can use any Android-based phone camera. On the hardware end, the camera broadcasts through a local area network (LAN) using a pre-set IP address. On the software end, the feed can be retrieved, processed, and displayed in real-time through any Python interpreter on a device connected on the same network. The spectrometer program depends on the following Python libraries:

- Numpy
- Matplotlib
- Scipy
- OpenCV
- Peakutils
- URLLib

The current features are as follows:

- Calibration information can be set within the program itself.
- Live feed of camera and corresponding intensity profile of a selected line scan region can be displayed in real-time on a computer with the required dependencies installed.
- Scale of relative intensity profile can be set by the initial camera exposure settings but is always normalized.

2 Theory

3 Setup

4 Program

4.1 `Spectrometer.__init__(calibrationLocation, calibrationWavelengths, lowerPix, upperPix, lowerBound, upperBound)`

Instantiates the Spectrometer object and takes the calibration arguments.

Table 1: Program initialization.

Parameters	
	<code>calibrationLocation</code> : <code>array_like</code> Pixel locations of the peaks of the calibration image.
	<code>calibrationWavelengths</code> : <code>array_like</code> Corresponding wavelengths of <code>calibrationLocation</code> .
	<code>lowerPix</code> : <code>int</code> Specifies pixel location of <code>lowerBound</code> (optional).
	<code>upperPix</code> : <code>int</code> Specifies pixel location of <code>lowerBound</code> (optional).
	<code>lowerBound</code> : <code>float</code> Specifies wavelength lower bound.
	<code>upperBound</code> : <code>float</code> Specifies wavelength upper bound.

4.2 `Spectrometer.plotCalibration()`

Plots the calibration curve and corresponding pixel-to-wavelength equation using linear regression.

4.3 `Spectrometer.LineScan.snapshot(image_name, peaks, window_length, polyorder)`Table 2: `LineScan.snapshot` arguments.

Parameters	
	<code>image_name</code> : str File name of locally-stored image.
	<code>peaks</code> : bool Sets whether peak points should be indicated on intensity profile.
	<code>window_length</code> : int Specifies window length of Savitsky-Golay filter.
	<code>polyorder</code> : int Specifies polynomial order of Savitsky-Golay filter.

4.4 `Spectrometer.LineScan.live(URL, show_peaks, window_length, polyorder)`Table 3: `LineScan.live` arguments.

Parameters	
	<code>URL</code> : str IP address of capturing device (Android-based phone camera only).
	<code>show_peaks</code> : bool Sets whether peak points should be indicated on intensity profile.
	<code>window_length</code> : int Specifies window length of Savitsky-Golay filter.
	<code>polyorder</code> : int Specifies polynomial order of Savitsky-Golay filter.

5 Demonstration**Appendix**

Source code:

<https://colab.research.google.com/drive/1VMUdZ9GGeLgUW5F7rmk0VZNeu9xxkdcU>.