EC552 Final Project: Team uDROP Generation

Team Members: Diana Arguijo, Benjamin Li, Poras Shroff, Mercedes Suazo

Description

uDrop-Generation is an application that uses image processing techniques to perform droplet detection, quantification of droplet diameter, and quantification of droplet generation rate within a microfluidic device. The goal of our project l is to improve the functionality of the application to handle variations in the droplet videos. These variations can include transparency, oils, as well as background noise. It is important to increase the accuracy and precision of the uDrop-Generation tool, as this will allow a more efficient lab process to be implemented, saving time and cost.

Software Component Functionality

FilterVideo Routine

The **FilterVideo** routine is responsible for applying the filter selected by the user to every frame of the image. This is done using the CV2 python module, which provides the functionality to splice the original image into its individual frames, stored as objects. Additionally, by passing in the frame, a user-specified thresholding algorithm, and threshold pixel value limits into the CV2 threshold API, each frame can be modified and then appended to the output video.

Edge Detection Threshold Modification Module

The **runAnalysis** function includes the edge detection threshold modification loop. If the average pixel value over all the frames in the video is lower than the minimum value defined in the module, the edge detection thresholds are lowered.

Edge Detection Fine Tuning Module

The **edge_fine_tuning** function performs the optional fine tuning of the edge detection thresholds. This module uses the first modification loop as a starting point for the thresholds and systematically changes the high and low values. After all the iterations the best performing combinations are printed in the terminal window, and the combination with the lowest standard deviation is shown in the GUI.

Special Instructions

Compilation of the code follows the same instructions as the original uDROP-Generation application. The major difference is that the startup command has been modified to introduce selection of the filter. Valid commands that have been added are shown below:

‘**python3 uDROP_Generation.py {filepath_to_video} filter**’

will apply the default filtering algorithm (Truncate) to the video located at **{filepath_to_video}.**

Example: ‘**python3 uDROP_Generation.py videos/clear.mov filter**’ will apply the truncating filter to the video located at the relative directory ‘**videos/clear.mov**’.

‘**python3 uDROP_Generation.py {filepath_to_video} filter {filter-type}**’
will apply the selected filter {filter-type} to the video located at **{filepath_to_video}.**
Valid options for {filter-type} are:

- ‘truncate’
- ‘binary’
- ‘tozero’

Example: ‘**python3 uDROP_Generation.py videos/clear.mov filter binary**’ will apply the binary filter to the video located at the relative directory ‘**videos/clear.mov**’.

‘**python3 uDROP_Generation.py {filepath_to_video} filter {filter-type} edge_fine_tune**’
will apply the selected filter {filter-type} to the video located at **{filepath_to_video}** and fine tune the edge detection thresholds.

It should be noted that just like the original uDROP_Generation application, not specifying **{filepath_to_video}** will use the frames from the previous video, which *can* be a filtered video if it was applied previously. A filter cannot be specified without already specifying **{filepath_to_video}**. This will result in unspecified behavior.