CS 513: Geospatial Vision & Virtualization

Assignment 1

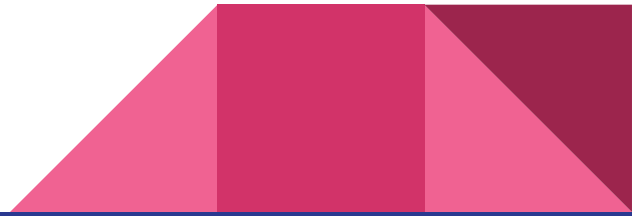
Automatic smear detection on the camera lens from given street view images

Group Members

- Ankita Acharya (A20436552)
- Rashi Dhande (A20396345)
- Neethi Kavadi (A20)
- Chetan Gupta (A20378854)

Agenda

- Introduction
- Previous Approaches
- Approach
- Functions Used
- Steps to Compile and Execute
- Intermediate Results
- Final Results
- References
- Contents of .zip folder



Introduction and Analysis

- We need to detect smear present on the camera lens through various image processing algorithms
- The smear appears mostly in all images present in the dataset and its location is constant throughout
- The remaining additional background appears to be the presence of noise
- The algorithms aim to process and filter out the noise present in the images
- The resulted image is noise free and the smear appears with its contour if it exists in the dataset
- Anaconda for Python 3.6 and OpenCV 3.1.0 is used to implement the algorithm

Previous Approaches - Brainstorming

- Background/Foreground detection
  - In this approach we were not able to detect edges as well as some smears in the images
- History variable
  - We put the image in the history, but that didn't work out, so then we used averaging image on sequence of images taken as input and smoothen using Gaussian Blur/Gaussian filter
- Background - Subtraction
  - We devised to try this algorithm to see whether we can put together several images and output a binary image of white and black smear. The unchanged constant in the series of images remain black, while the white outlines are drawn to indicate the constantly changing portions by which we can use to detect the unchanging smear in the picture. We were forced to devise another algorithm as this was only working for video inputs and not images.

Approach

- Initially the dataset of all the images are extracted from specific directory
- All the images are parsed and scaled down
- The laplacian gradient image is generated from each image and averaged together into one image
- Then the detection of smear is processed onto the average of all the images captured and is blurred for easier contouring
- Remaining noise is removed via dilation
- Epsilon and threshold values are fixed for edge detection
- Contour for the smear are mapped

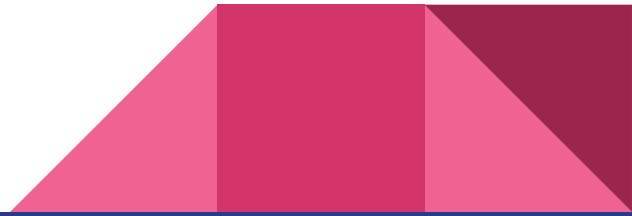
Note: due to dilation, the contour mapped is an approximate estimate, not exact

Functions Used

- **GaussianBlur()**
  - This function converts the source image with the specified Gaussian kernel
- **cvtColor()**
  - This function is used to convert images from one color space to another
- **threshold_adaptive()**
  - This function is used to convert the Grayscale image to Binary image
- **Canny()**
  - Detection algorithm to find the edges in the Binary image
- **findContours()**
  - This function is used to detect the contours of the edges
- **drawContours()**
  - This function is used to mark the detected smear in the image.
- **Dilate**
  - So it increases the white region in the image or size of foreground object increases

Steps to Compile and Execute

1. Install python
2. Install packages as needed [see top of .py file for reference]
3. To detect smear of lens used: simply update the ‘path’ variable in .py file to the location of your sample drive and run .py file - be careful to follow format of path similarly to given code [Some paths include ‘cam_’ and some don’t
    - NOTE: Resizing all the images may take a while -
    - Outputs and Intermediate results will be made in directories inside your sample_driver following execution



Intermediate Results

- Average Image with Smear for Cam 3:

Intermediate Results

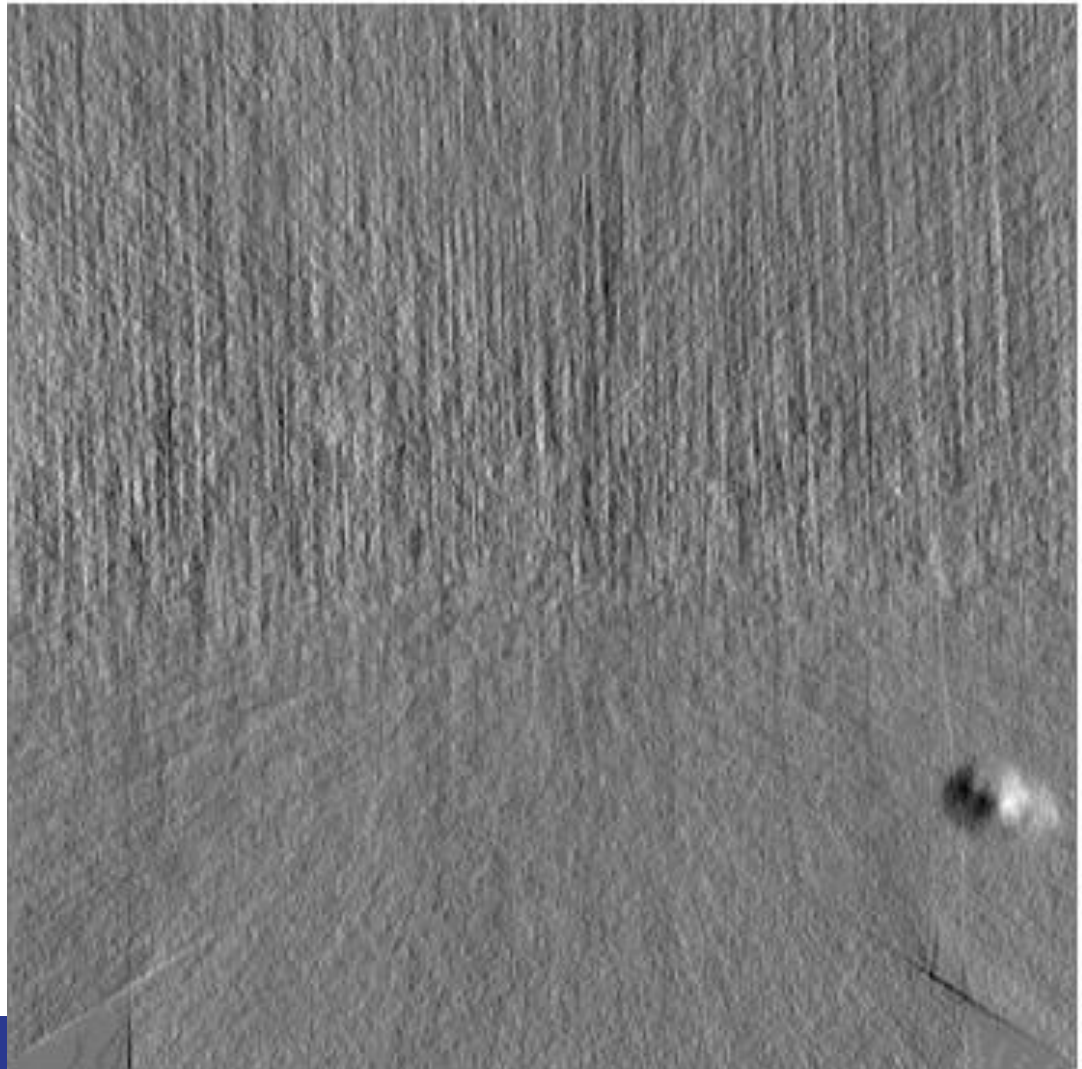
- Average Image without Smear for Cam 5:

Intermediate Results (contd…)

- Canny Edge Detected Image:
- For Cam 3
- With dilation

Intermediate Results (contd…)

- Laplacian Gradient:

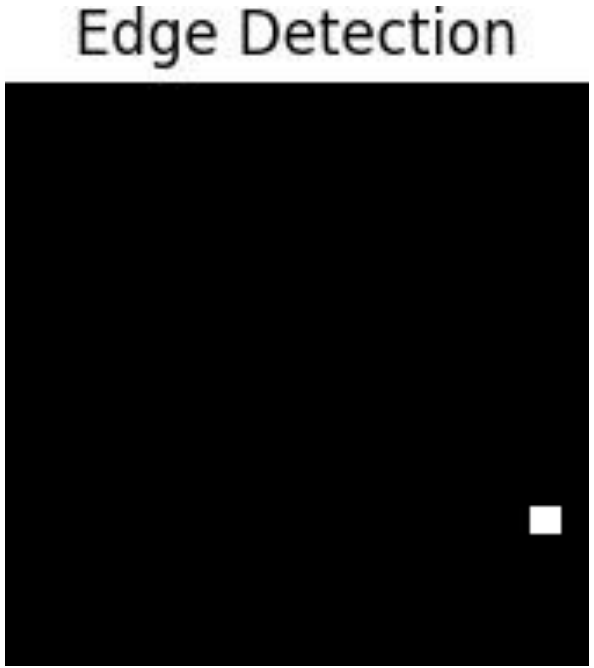


Final Results

Original image with smear detection

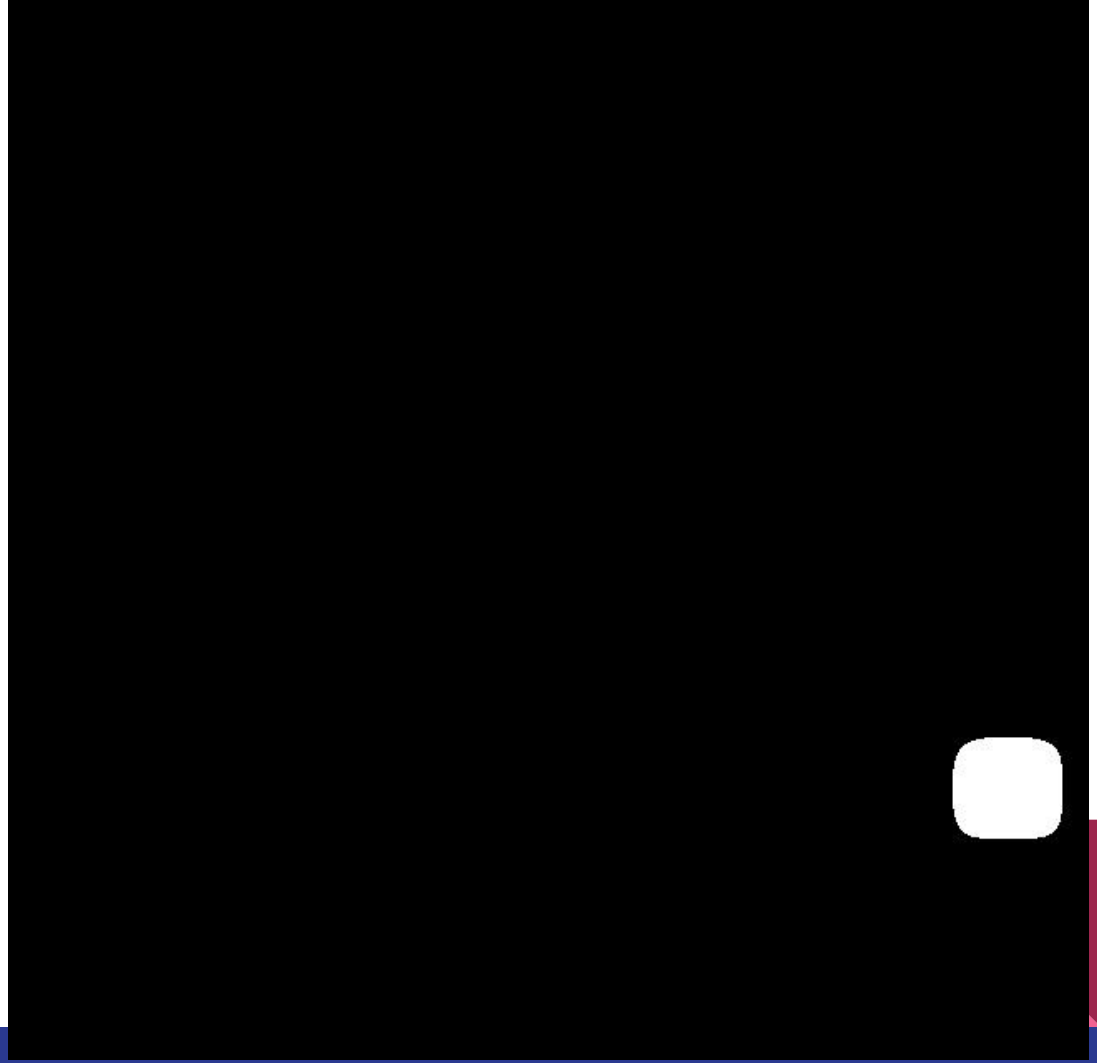
Final Results

- Smear Contour Image:



Final Results (contd...)

- Masked Image:



Contents of zip folder

- Source code in Python : **smear_final.py**
- Average Images
- Canny Edge detected Image
- Laplacian Gradient
- Smear contour detection on original random photo for each cam
- Masked Image
- Presentation slides

References

- Removing Image Artifacts Due to Dirty Camera Lenses and Thin Occluders," J. Gu, R. Ramamoorthi, P.N. Belhumeur and S.K. Nayar, ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) - http://www.cs.columbia.edu/CAVE/publications/pdfs/Gu_SIGGRAPH_Asia_09.pdf
- https://forums.ni.com/t5/Machine-Vision/Detect-Dirt-on-a-Camera-Image-with-Vision-Assistant/td-p/1999221
- OpenCV documentation - http://docs.opencv.org/2.4/modules/refman.html
- http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html

THANK YOU !