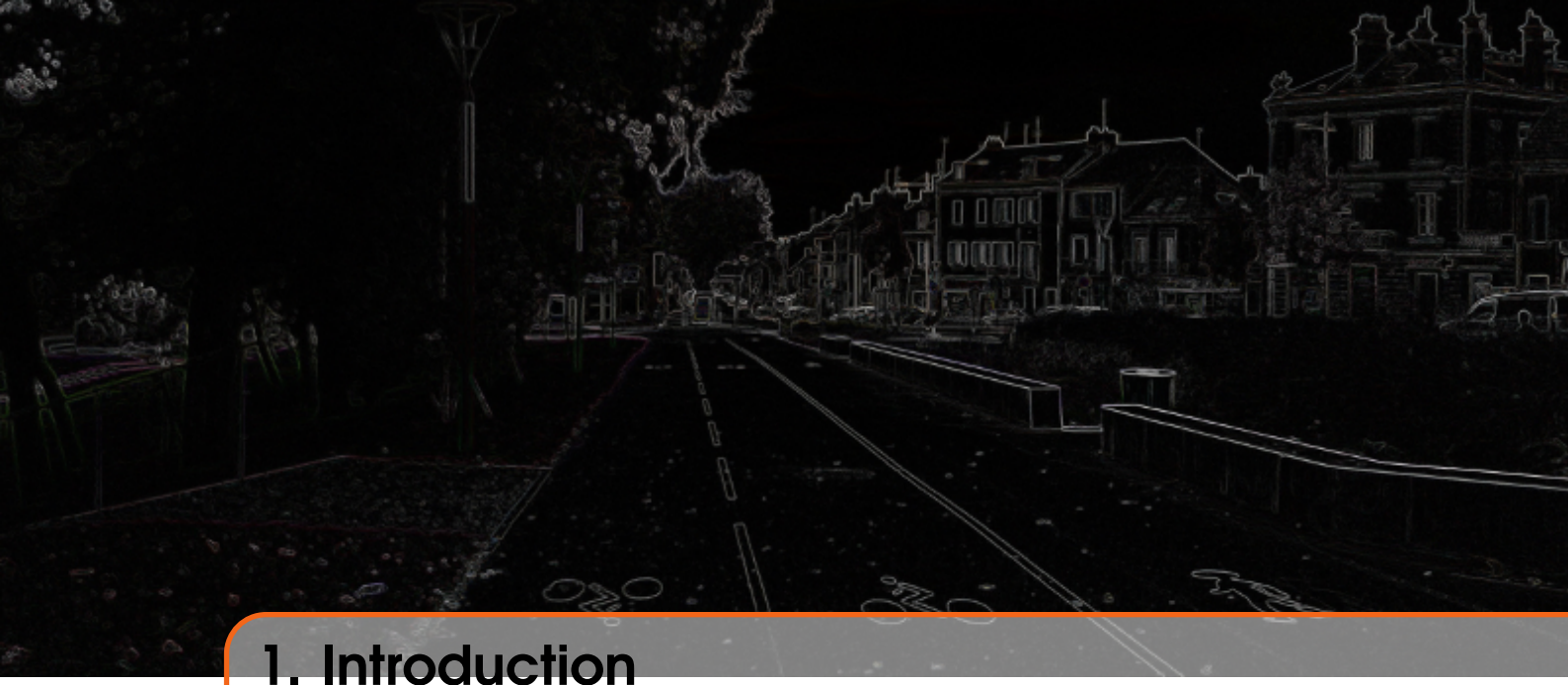
User Manual for Computer Vision toolbox

Developed with Matlab

Sandeep Manandhar
MSCV6
Université de Bourgogne

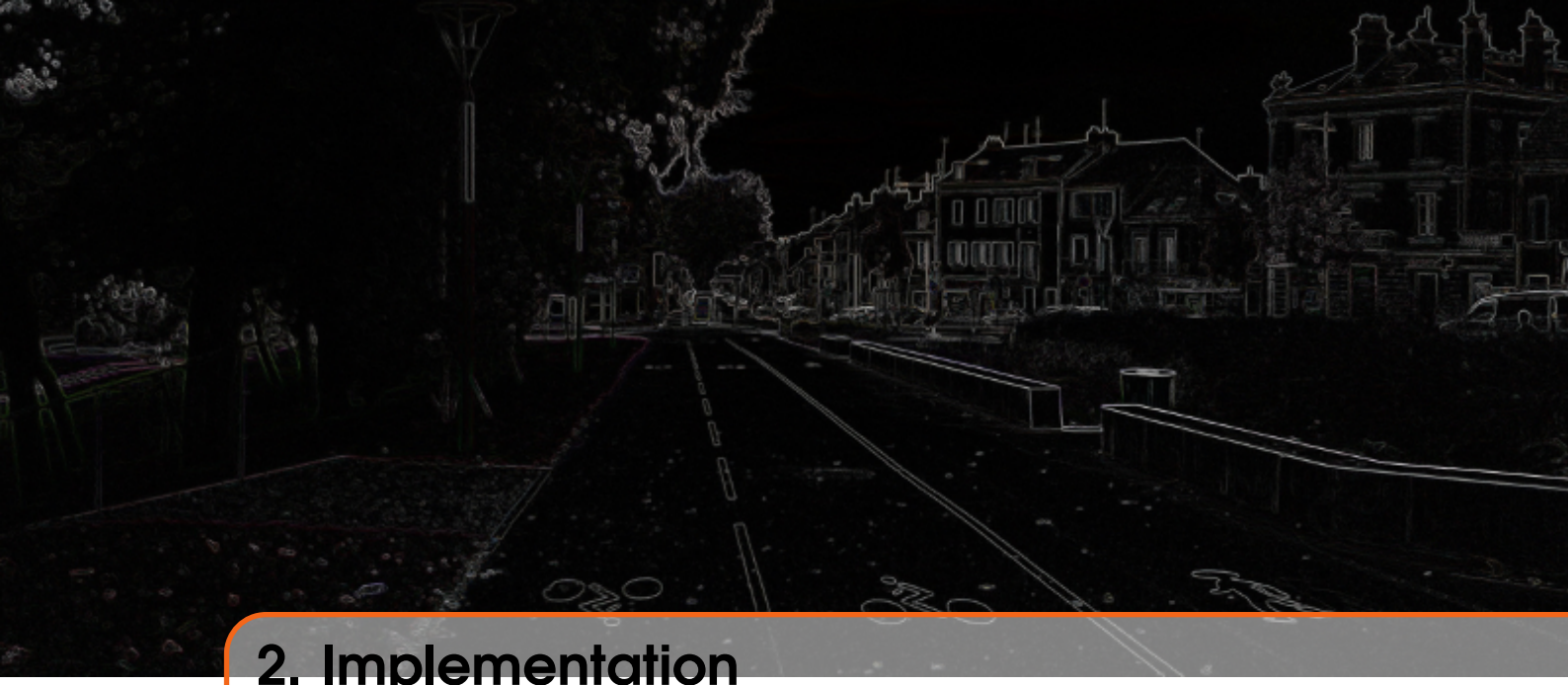
Contents

Part One



1. Introduction

Learning image processing and pattern recognition and practicing the algorithms in the domain of computer vision requires proper tool box. The widely used Intel's OpenCV library and Mathworks Matlab requires one to have a considerable amount of programming languages. This project aims at developing a software sufficient to visualize the basic algorithms of computer vision. The developed software has an user friendly interface that curtains all the hasty and tedious codes and lets user to play around with the various parameters to understand the text book algorithms.



2. Implementation

A large set of computer vision algorithms have been implemented ranging from basics to some of the computationally involved ones. Matlab's Image Processing toolbox has been used to develop this software. The list of the algorithms implemented are:

- Color Conversion
- Brightness and Contrast control
- Histogram equalization
- ROI selection and Watermarking(logo)
- Noise addition
- Global and Local thresholding
- Mean, Median, Gaussian filters and custom filters
- Canny Edges, Sobel edges, Hough lines/circles and Contours
- Morphological operations : Dilate, Erode, Open, Close, Gradient, Top hat/black hat
- SIFT, SURF, FAST, MSER features
- Fundamental Matrix, Homography, Epipolar lines
- Camera Calibration with chessboard pattern

2.1 User Interface

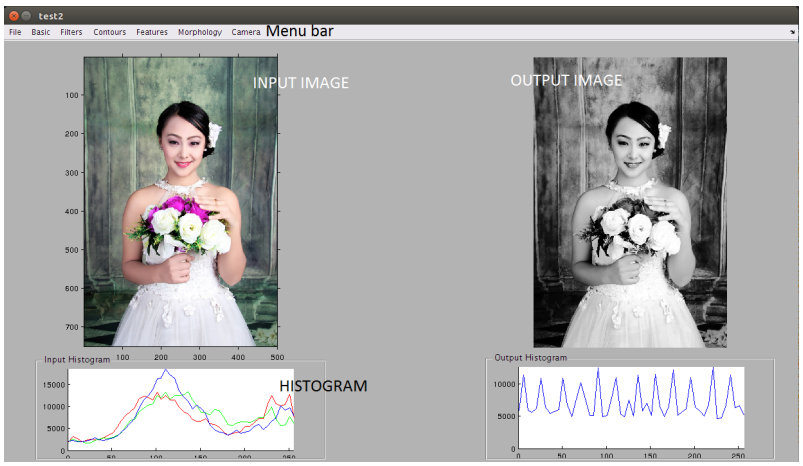


Figure 2.1.1: *User Interface*

The figure 2.1.1 shows the general overview of the user interface. The major components as depicted are:

1. Input Image display
2. Output Image display
3. Histogram
4. Menu bar

2.1.1 Workspace

The workspace can take up to the whole space of the user interface. A user can open only one image. All the matlab supported images are permitted. **File** menu provides opening and saving of an image as depicted in 2.1.2.

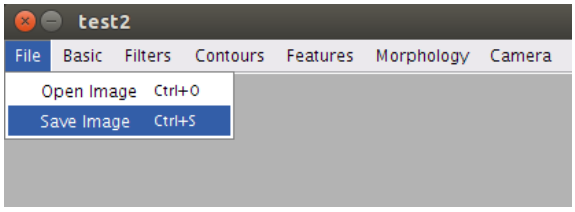


Figure 2.1.2: *file menu*

2.1.2 Histogram Visualization

Histogram visualization is available for images. Single or triple channel images can have their histogram visualized. Loading an image makes the histogram appear on the bottom part of the image. Some the process like feature detection do not create histogram.

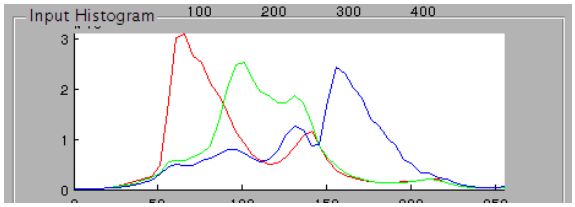
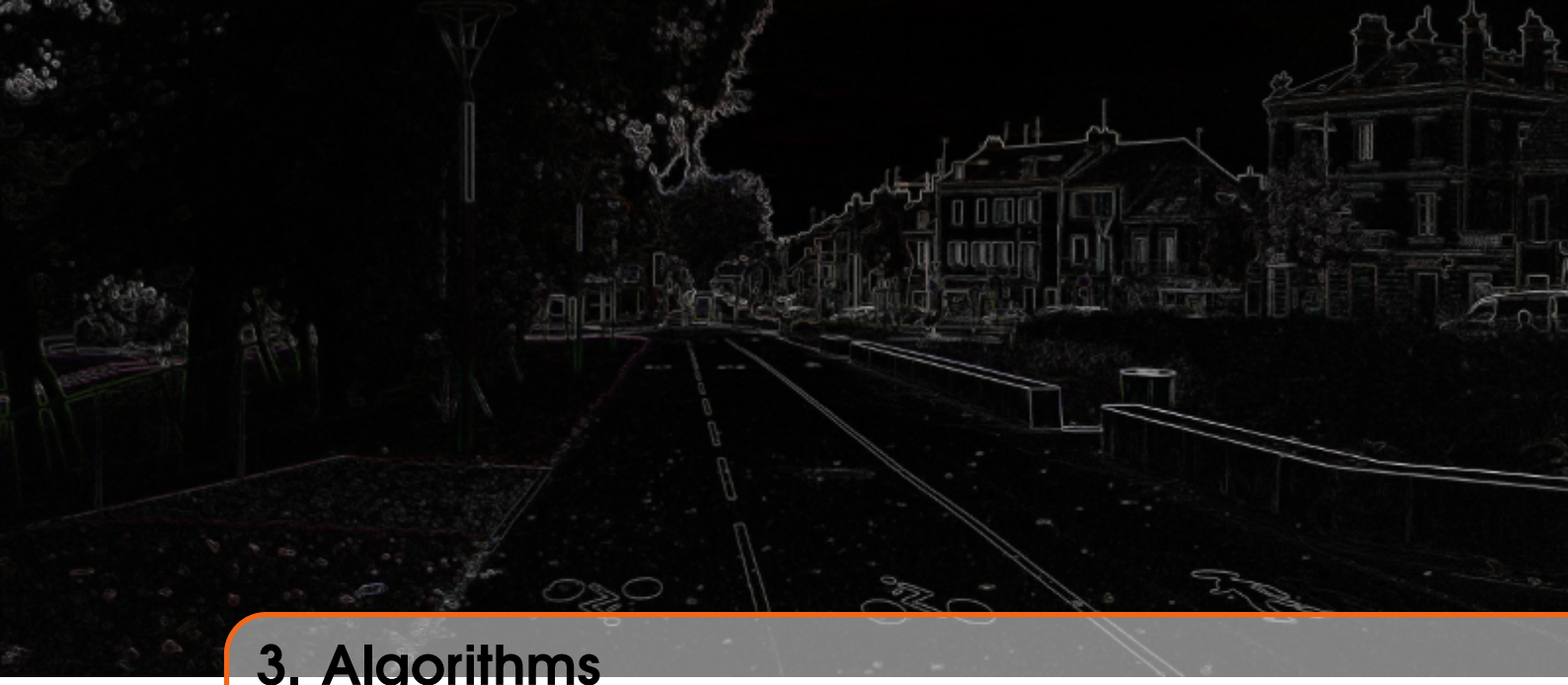


Figure 2.1.3: *Histogram Window*



3. Algorithms

3.1 Basic

3.1.1 Color Space

Color space conversion for RGB to grayscale, YCbCr, NTSC, HSV, and inverted space are available. It is to be noted that, none of them can be reverted back to RGB space in this software.

3.1.2 Brightness and Contrast

The Brightness and Contrast menu provides two slider for the user to play with. The image will be updated along with the slider movement. It is to be noted that the system can only apply these changes to an original image and not to the processed ones, **for now**.

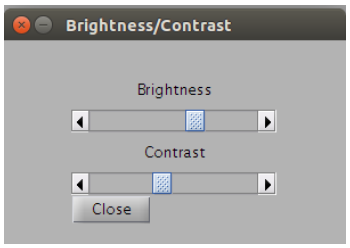


Figure 3.1.1: *Brightness and Contrast*

3.1.3 Add Logo

To watermark or place a logo in the image, select **Basic->Add logo** then select a file. The user should create a rectangular area in input image by dragging mouse across the input panel. Then the rectangular region is to be double clicked. A dialog will pop up to set the weight for the logo.

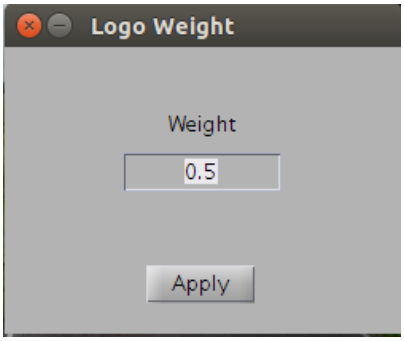


Figure 3.1.2: *weight of logo*

Figure 3.1.3: *logo added*

3.1.4 Local Thresholding

Local thresholding can be applied to a grayscale image **Basic->Local thresholding**. The program asks user to set kernel size and offset level.

3.1.5 Histogram equalization

Histogram equalization is available for grayscale images only, that can be selected as **Basic->Histogram Equalization**.

3.1.6 Add noise

Available noises to be added to the image are Gaussian, Poisson, Salt and Pepper, and Speckle.

3.2 Filters

3.2.1 Pre-made filters

Average, Median , Gaussian blurs, Sharpen, Sobel and Laplacian filters are available as ready-made filters.

3.2.2 Custom filters

To make things more interactive, a custom kernel wizard has been provided where user can create his/her own kernel of any size. To to **Filters->Custom** in the menu to select it. The following window will appear where customization is possible.

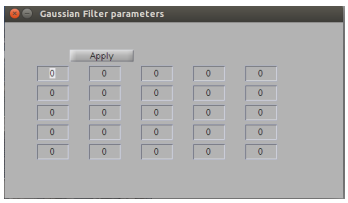


Figure 3.2.1: *kernel size of 5*

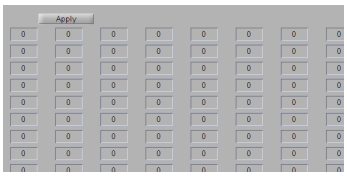


Figure 3.2.2: *User can select a huge kernel*

3.2.3 Contours

Contours can be selected as **Contours->showCotours**.

3.3 Features

3.3.1 Canny

Canny edges can be found by selecting **Features->Canny**. A dialog box will appear asking for various parameters.

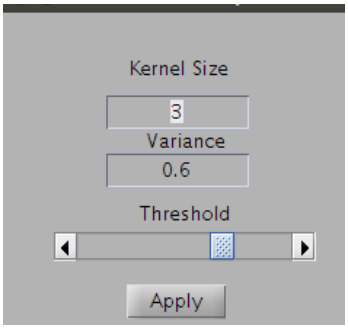


Figure 3.3.1: *canny parameters*

3.3.2 Hough

Similarly, hough circles, lines can be detected from the feature menu. Clicking on each of them will prompt for the various paramters.

3.3.3 FAST/SURF/Harris

FAST, **SURF** and **HARRIS** corners are available for the detection. **Show matches** option will show matches between two or more images.

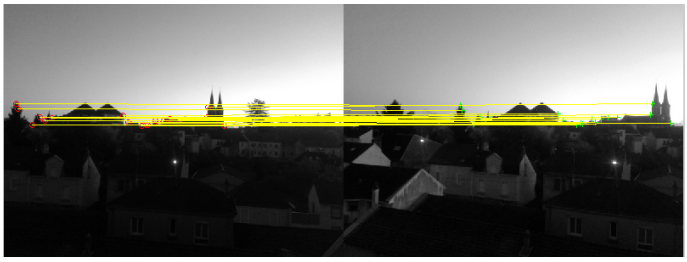


Figure 3.3.2: *matches with FAST features*

3.3.4 Morphology

Morphology A basic set of morphological operators has been provided. The available operations with their respective console commands are:

- dilate
- erode
- open
- close

The parameters for these operations are available for tuning at **Morphological->Choose**.

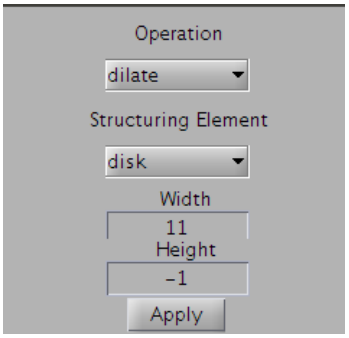


Figure 3.3.3: *Morphological gradient with tuned parameters*

3.3.5 Calibration

The software has been assisted by `OCAMCALIB` toolbox for the calibration. The calibration routine can be performed by **Camera->calibrate**.

3.3.6 Fundamental Matrix calculation

To calculate fundamental matrix, choose an input image and select **Camera->Fundamental matrix**. Then select the second image for the secondary plane. The calculation would result in the fundamental matrix formation as well as the image with epipolar lines in both images.

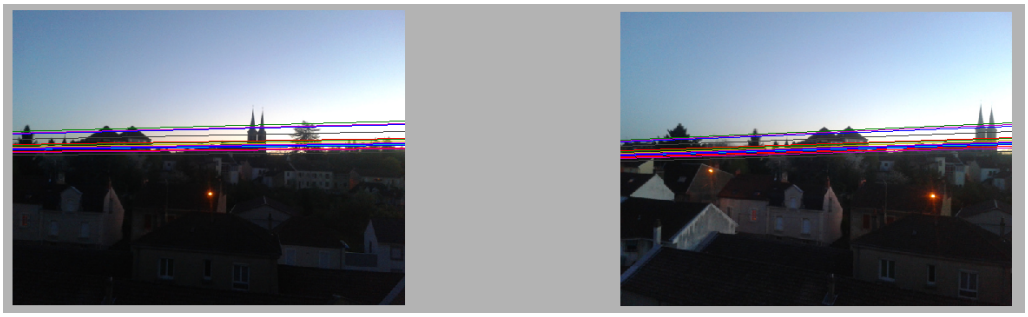


Figure 3.3.4: *Epipolar lines from fundamental matrix*

3.3.7 Homography

Homography The calculation of homography has been kept simple. Mosaic image is made based on homography.Choose an input image and select **Camera->Mosaic Homography**. Then select the second image and the output image will be an stitched image.

Part Two

3.4 BONUS please!!

Nothing much has been done beyond what was asked for. I would like to mention the local thresholding was written from scratch using integral image technique to boost up the calculation.

Bibliography

References

- http://fr.mathworks.com/matlabcentral/, Matlab Online Documentation and help website, last accessed at

- Robert Laganière, “OpenCV 2 Computer Vision Application Programming Cookbook". 2011

Index