

User Manual for Computer Vision toolbox

Developed with C++/OpenCV

**Sandeep Manandhar
MSCV6**

Université de Bourgogne



Contents

I	Part One	
1	Introduction	7
2	Implementation	9
2.1	User Interface	10
2.1.1	Workspace	10
2.1.2	Console	11
2.1.3	Images and Videos	11
2.1.4	Histogram Visualization	12
2.1.5	Miscellaneous	12
3	Algorithms	13
3.1	Image	13
3.1.1	Color Space	13
3.1.2	Brightness and Contrast	13
3.1.3	Watermarking	13
3.1.4	Crop	14
3.1.5	Histogram equalization	14
3.1.6	Up/Down sampling	14
3.2	Filters	15
3.2.1	Pre-made filters	15
3.2.2	Custom filters	15
3.3	Contours	15
3.3.1	Canny	15
3.3.2	Sobel	15

3.3.3	Laplacian	15
3.3.4	Hough line and circle	16
3.3.5	Options and Settings	17
3.3.6	Morphology	17
3.3.7	Features	18
3.3.8	Calibration	18
3.3.9	Solve Perspective to Point	19
	Solve Perspective to Point	19
3.3.10	Fundamental Matrix calculation	19
3.3.11	Homography	20
3.3.12	Face detection	21

II

Part Two

Anexes	25
3.4 Can I have some BONUS points!!!	25
3.5 List of console commands	26
3.5.1 Known bugs	27
Bibliography	29
References	29
Index Alphabetical	31



Part One

1	Introduction	7
2	Implementation	9
2.1	User Interface	
3	Algorithms	13
3.1	Image	
3.2	Filters	
3.3	Contours	



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. The OpenCV API, being robust and efficient, has been thoroughly utilized to do so. 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
- Camera Pose estimation
- Face/eyes detection

2.1 User Interface

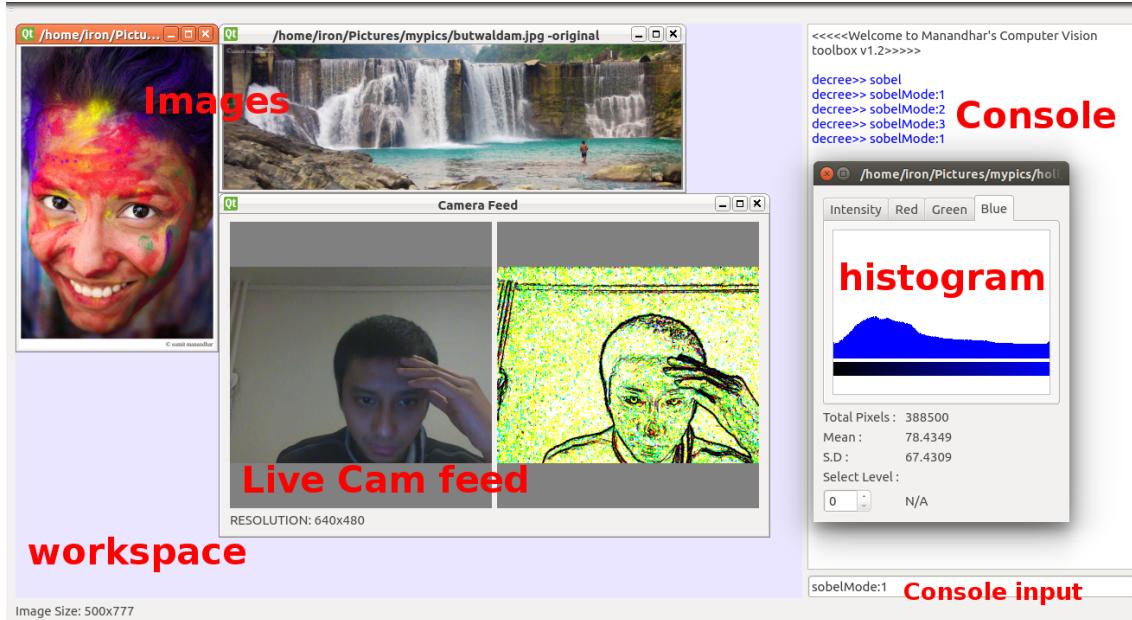


Figure 2.1.1: User Interface

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

- Workspace (a) Images
 (b) Live Camera feed
 (c) Offline video
 - Console (a) Command Input
 (b) Command output
 - Histogram (a) 3-Channel histogram
 (b) Statistics-Mean and Standard Deviation
 - Miscellaneous (a) Settings and wizards
- ...

2.1.1 Workspace

The workspace can take up to the whole space of the user interface. The user can opt for opening multiple images, a single live camera feed or a single offline video. Currently supported images are of type **JPEG**, **PNG** and **BMP** and the supported video format is **AVI**.

Opening any of these subjects are pretty straight forward. **File** menu provides these options as depicted in 2.1.4.

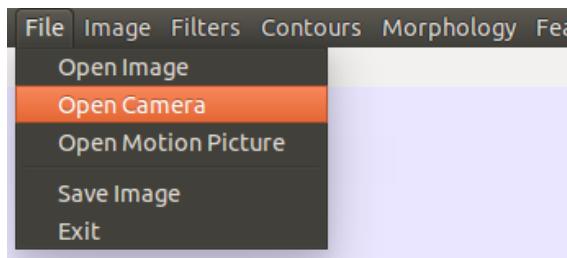


Figure 2.1.2: file menu

2.1.2 Console

The console provides a minimal interface to call various functions without using the menu bar. Press **F9** on keyboard to toggle the console window which will be active on the left side of the interface 2.1.1. The available commands are listed below. It is to be noted that calling functions and setting parameters are done with separated commands.

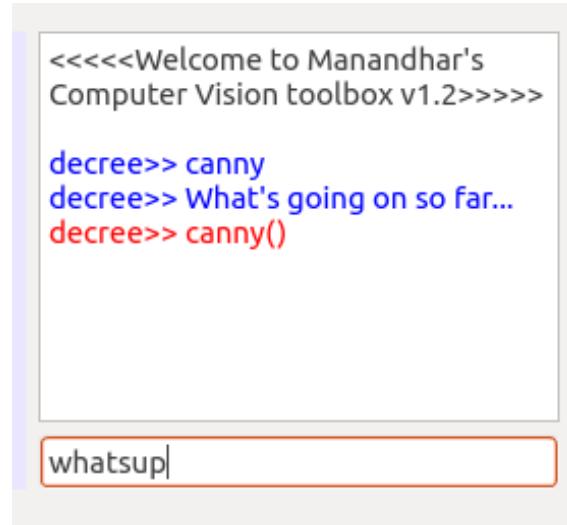


Figure 2.1.3: file menu

2.1.3 Images and Videos

Images open as a single window. All of the processes are applied to the same window. To compare with the original, one can use **Windows->Original**.

In case of videos, be it live or offline, all the processes are applied to a separate frame. The video window will have two preview sub-windows; right one being the input stream and the left one being the output stream 2.1.5

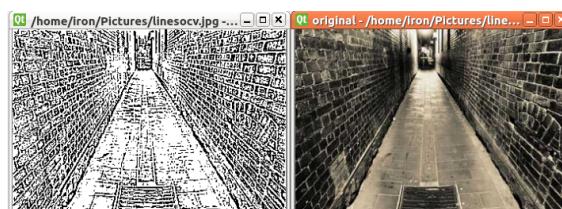


Figure 2.1.4: Processed image(Left) and Original image(Right)

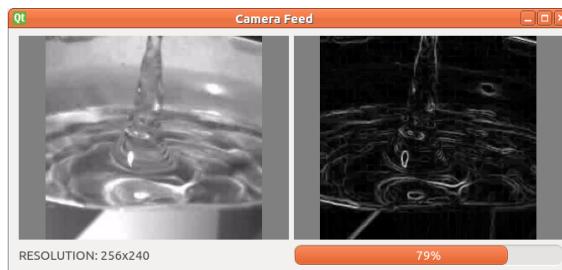


Figure 2.1.5: Input frame(Left), Processed frame(Right), Frame details(Bottom)

2.1.4 Histogram Visualization

Histogram visualization is only available for images. Single or triple channel images can have their histogram visualized. Intensity, Red, Green and Blue tabs are available to view the histograms. Statistics: Mean and Standard deviation, are available on the bottom panel.

The histogram window will be available be **Windows->Histogram**. The visibility of the window can be toggled from the same menu. The window will appear as the floating window besides the main interface.

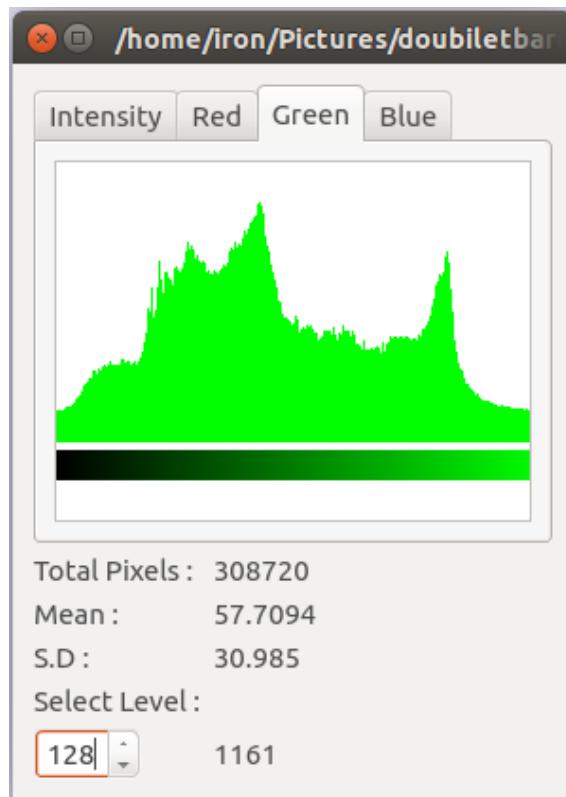


Figure 2.1.6: Histogram Window

2.1.5 Miscellaneous

Additional wizards and windows accompany the interface for tuning parameters of various algorithms. These tools will be discussed during the description of the algorithms in coming sections.

3. Algorithms

3.1 Image

3.1.1 Color Space

Color space conversion for RGB to grayscale, YCbCr, YUV, HSV, Lab and HLS 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**.

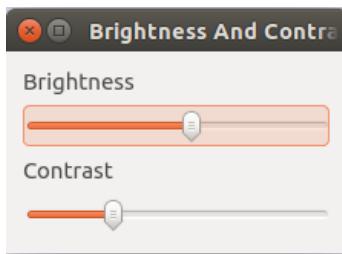


Figure 3.1.1: Brightness and Contrast

3.1.3 Watermarking

To watermark or place a logo in the image, user defines the rectangular region on the image by dragging the mouse. A green rectangle will appear as the mouse is being dragged. After finishing the selection of region, the user should select **Image->Place Watermark->watermark it!!** from the menu. Then a file selection dialog will appear. The user needs to select the logo image to be place in the previously defined region. The image will be re-scaled according to the region defined. To set the transparency, select **Image->Place Watermark->Transparency** for the menu. A horizontal sliding widget will appear where the transparency degree can be selected.

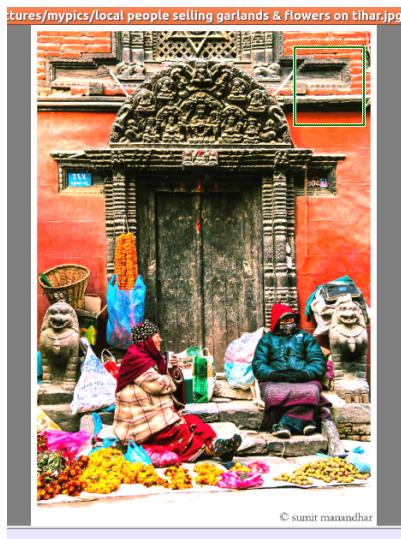


Figure 3.1.2: ROI defined in the image

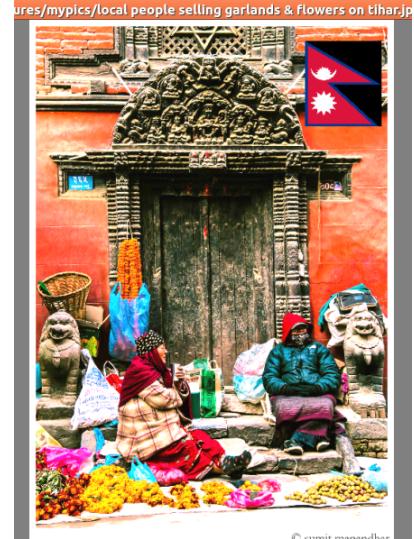


Figure 3.1.3: Flag of Nepal as a logo

3.1.4 Crop

Cropping images follow the same initial steps as watermarking. Select a region of interest by dragging the mouse and select **Image->Crop** from the menu bar. The cropped image will be displayed in a new image window.

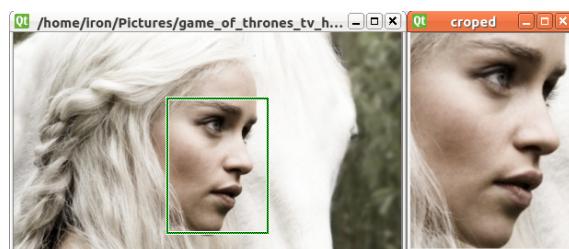


Figure 3.1.4: cropping an image

3.1.5 Histogram equalization

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

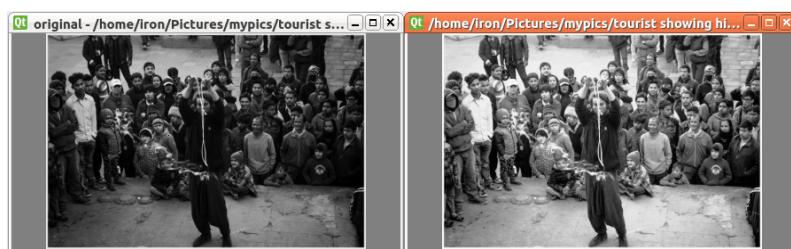


Figure 3.1.5: equalized image(on left)

3.1.6 Up/Down sampling

Images can be sampled up or down by 2x or 1/2x scales respectively. Select **Image->Up Sample** and **Image->Down Sample**.

3.2 Filters

3.2.1 Pre-made filters

Average, Median and Gaussian blurs are available as ready-made filters. Addition of salt and pepper noise is also available. To select any of these filters go to **Filters** in the menu.

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 go to **Filters->Custom** in the menu to select it. The following window will appear where customization is possible. Pressing **Make it happen** button will immediately apply the filter to the active image. The wizard is available for the videos as well.

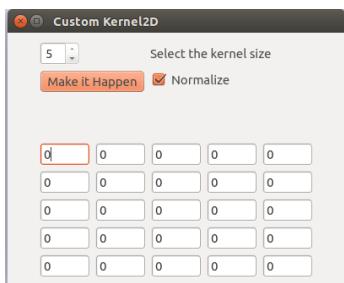


Figure 3.2.1: kernel size of 5

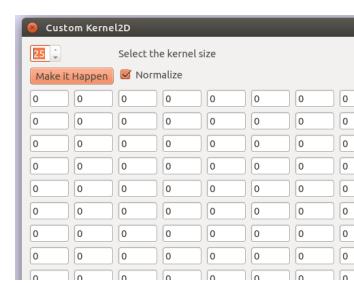


Figure 3.2.2: User can select a huge kernel

3.3 Contours

3.3.1 Canny

Canny edges can be detected for images and videos as well. The thresholds can be defined from the settings as shown in 3.3.4. For a video, this function can be called from the console command by **canny**.

3.3.2 Sobel

Sobel edges can be detected for the subjects by selecting **Contours->sobel** from the menu and choosing any of the modes. These modes display the Magnitude, Orientation and Sobel map of a given image. For a video, this function can be called from the console command by **sobel**. By default, this command will output the magnitude map of the video frame. The mode selection in the console is set by these commands.

- sobel:1 - magnitude map
- sobel:2 - orientation map
- sobel:3 - sobel image

Other settings can be made from the **Contours->Options..** 3.3.4.

3.3.3 Laplacian

The laplacian filter can be selected from **Contours->Laplacian**. Similarly, the console command is defined as **laplacian**. The kernel size can be fixed from 3.3.4.

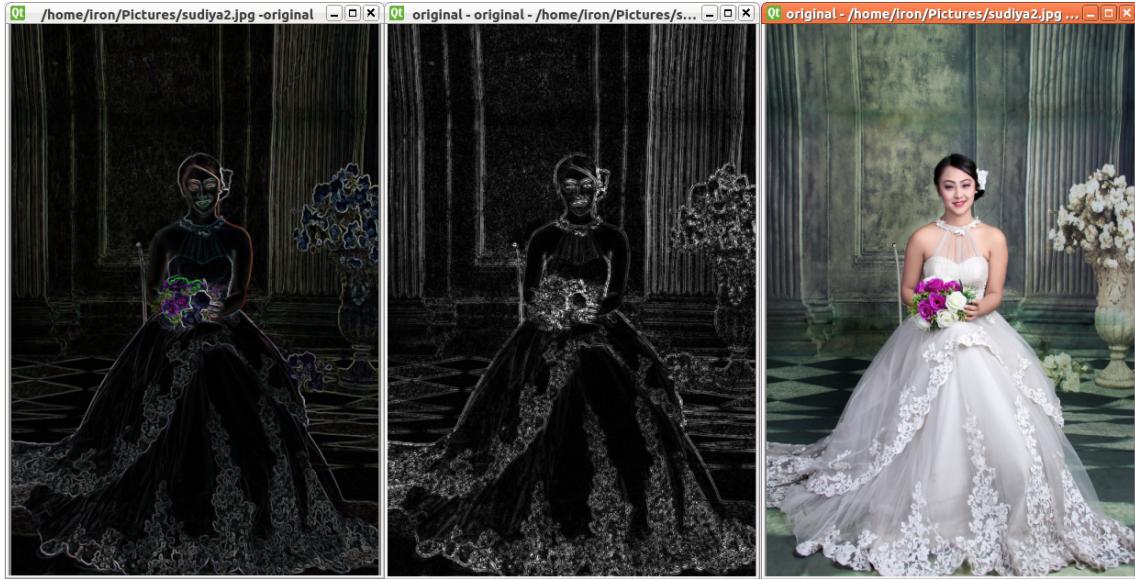


Figure 3.3.1: Sobel(left), Laplacian(middle), original(right)

3.3.4 Hough line and circle

Hough lines and circles can be detected by selecting **Contours->Hough lines** and **Contours->Hough circle**. Probabilistic Hough lines can be selected by **Contours->HoughPLines**. The console commands for line detection using these two variants are **hough** and **houghPLines**. It is to be noted that, hough circles are not available from the console.

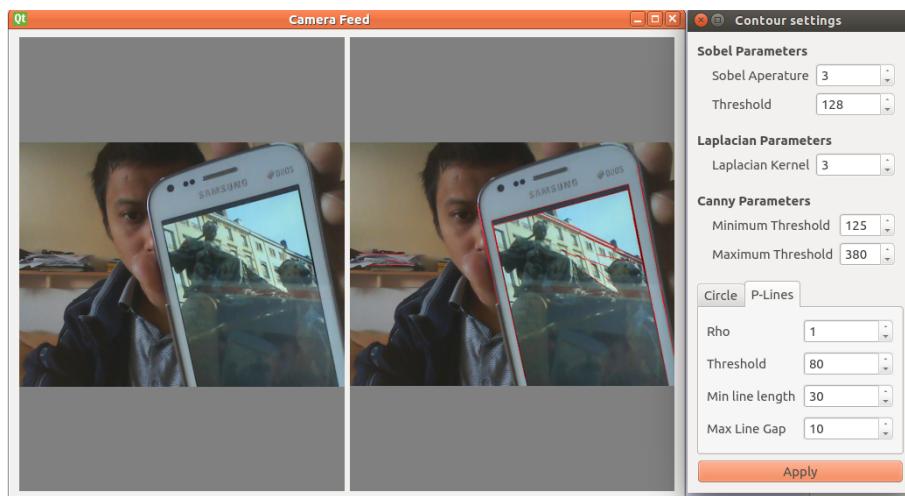


Figure 3.3.2: Hough Lines detected on a live feed



Figure 3.3.3: In this image: Circles of Barracuda swarm

3.3.5 Options and Settings

The setting for this section is available at **Contours->Options..**

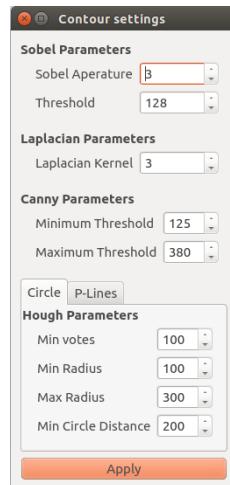


Figure 3.3.4: Parameters for edges and contours

3.3.6 Morphology

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

- dilate - morph_dilate
- erode - morph_erode
- open - morph_open
- close - morph_close
- gradient - morph_gradient
- top hat - morph_tophat
- black hat - morph_blackhat

The parameters for the operation are available for tuning in the options menu.

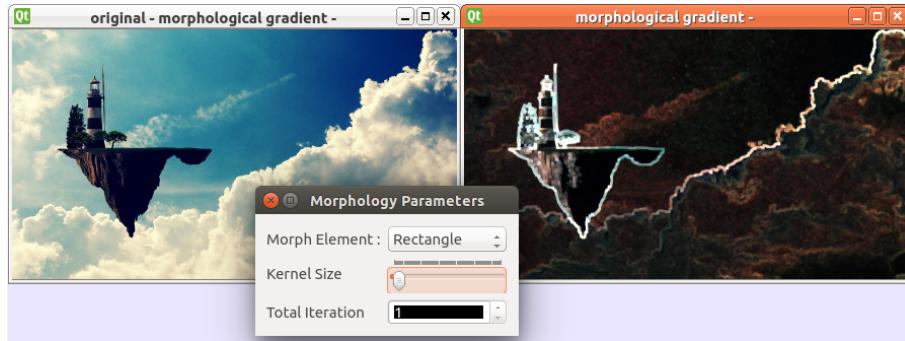


Figure 3.3.5: Morphological gradient with tuned parameters

3.3.7 Features

FAST, SURF, SIFT, MSER and Harris features are available for detection. These features can be detected in a live video feed but it is advised that SIFT not be used with the live video as it is computationally extensive.

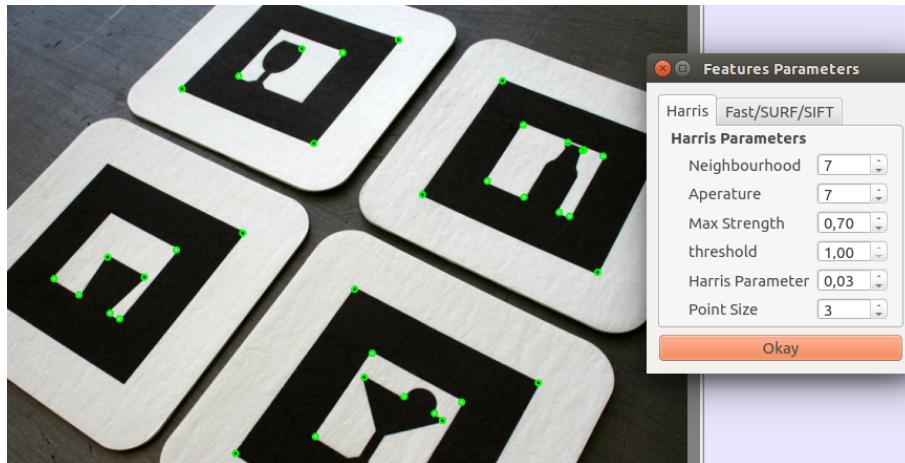


Figure 3.3.6: Harris corners

3.3.8 Calibration

The software provides one with a camera calibration tool box. The calibration requires a plane of chess board pattern. The calibration is done with the camera feed. No offline images can be processed for now.

The following steps should be taken to calibrate a camera:

- Open calibration wizard from **Camera Geometry>Calibrate** menu.
- Set the size of the board.
- Set the number of frames to calibrate with.(more than 7)
- Open Camera from the menu bar
- Press **T** on keyboard or select calibration from the menu.
- Put the chessboard pattern around the camera.
- The captured pattern will be displayed on left review widget. The show image will let user to see it maximized.
- After required number of frames have been detected, the progress bar will be filled and the user should select Calibrate button on the top left panel.
- The calibration will end with some error which will also be shown in the left panel. Try to make it as close as possible to zero.

- Save function is available to save the parameters of the camera.



Figure 3.3.7: Calibration wizard



Figure 3.3.8: Chessboard Pattern detected

3.3.9 Solve Perspective to Point

The calibration wizard has an added functionality to solve for perspective to point transformation. Put differently, camera pose can be estimated with the wizard. For this, it is necessary to have intrinsic parameters in hand. The following steps are required to do so:

- Load configuration for **File->Load Config**
- Select **Operation->SolvePnP**
- Choose a camera frame that has been saved in memory before.
- Choose the location to save the camera pose matrix.

```
SVAML:1.0
total Frames: 8
distortion Coefficients: !opencv-matrix
    rows: 5
    cols: 1
    dt: d
    data: [-1.6010245605555073e-01, 9.2649735573690384e-01,
           -5.492695357740436e-03, -1.5241804707504566e-03,
           -1.54583636389600793e+00]
```

Figure 3.3.9: Saved intrinsics

```
SVAML:1.0
Rotation: !opencv-matrix
    rows: 3
    cols: 3
    dt: d
    data: [ 9.4569949e+00, -4.95511e-01, -1.03012315916964e-01,
           2.37457920318205e-01, 8.132533889042475e-01, -5.748083992446425e-01,
           1.568799999842420e+01, 5.76234545959762e+01, 8.020512039515134e-01]
```

Figure 3.3.10: Saved Camera Pose

3.3.10 Fundamental Matrix calculation

The calculation of fundamental is done with the help of surf features. The following steps should be taken to calculate the matrix:

- Open an Image.
- Select **Camera Geometry->Options..**
- Select among 7-point, 8-point and RANSAC
- Tune the distance to epilines, nearest neighbour ratio
- Select **Camera Geometry->Fundamental Matrix**
- Select an image to create the transformation.

The calculation would result in the fundamental matrix formation as well as the image with epipolar lines in both images.

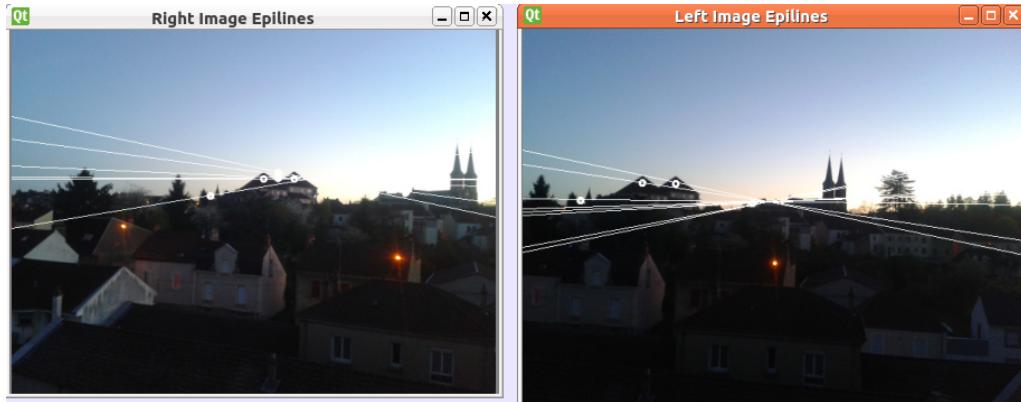


Figure 3.3.11: Epipolar lines from fundamental matrix

3.3.11 Homography

The calculation of homography has been kept simple. The following few steps should be taken to do so:

- Open an Image.
- Select **Camera Geometry->Homography**
- Select an image to create the transformation.

The result would be a stitched version of two images with found homography.

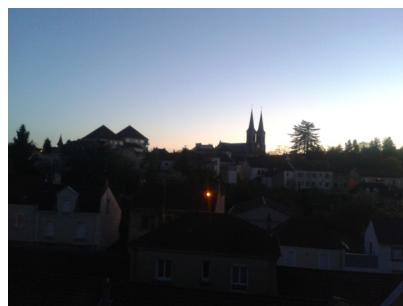


Figure 3.3.12: In this image: Le Creusot Chruch

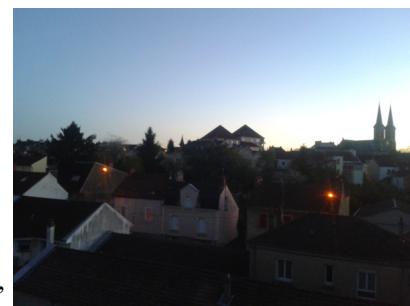


Figure 3.3.13: Another image of the scene

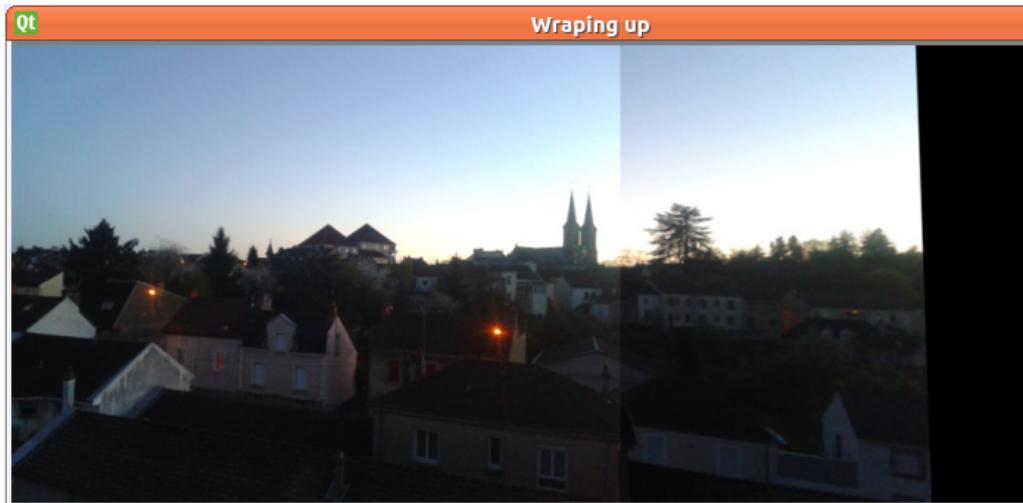


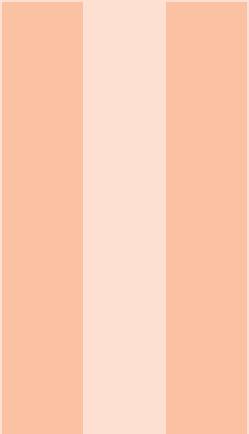
Figure 3.3.14: Stitching two images by Homography Matrix

3.3.12 Face detection

The tool box provides a face detection module. The function can be used across images and videos. Currently face and eyes detection are available. It is to be noted that, eyes will be detected only after the face has been detected. This was done so to minimize the search area for the eyes. Select **Learning->Detect human Faces**.



Figure 3.3.15: Face detected in the blue rectangle



Part Two

Anexes	25
3.4 Can I have some BONUS points!!!	
3.5 List of console commands	
Bibliography	29
References	
Index Alphabetical	31



Anexes

3.4 Can I have some BONUS points!!!

This section lists the features that have been added by the programmer himself on his free will with the hope that the generous supervisor might help him boost his score in the overall grade.

- Multi-window Interface based on OpenGL. refer 2.1.1
- Console to dispatch command based on function names. refer 2.1.3
- Local thresholding written from the scratch using basic OPENCV functions
- Make Border, Crop image, up/down sampling, interactive selection tool for Logo placement
- Customizable kernel for convolution of desired size. refer 3.2.1
- Camera calibration wizard using video frames every 3-seconds with customizable frame number and board-size
- Solve Projection -n-point to find camera pose
- Saving Extrinsic and Intrinsic parameters and Camera Pose to a file
- Pipeline base Video processing to stack one process over another
- Face and e detection
- **Working on adding Augmented reality module.** Hopefully by the end of the semester.

3.5 List of console commands

The following list of commands are available in the console for this version of the software.

Commands	
gray	Converts image to grayscale
YcrCb	Converts image to YCbCr
HSV	Converts image to HSV
HLS	Converts image to HLS
Lab	Converts image to Lab
Luv	Converts image to Luv
flipH	Flips image around x-axis
flipV	Flips image around y-axis
flipL	Flips image around both axes
filter	Applies custom kernel to image
saltNpepper	Add salt and pepper noise
canny	Detects canny edges
sobel	Finds sobel magnitude or orientation or edges
blob	Finds contours or bounding rect or convex hull or bounding circle
laplacian	Perform laplacian filtering
hough	Find hough lines or circle
houghPLines	Find hough lines based on probability
fast	Find FAST features
surf	Find SURF features
sift	Find SIFT features
mser	Find MSER features
harris	Find HARRIS corners
morph_dilate	Perform dilation
morph_erosion	Perform erosion
morph_open	Perform opening
morph_close	Perform closing
morph_gradient	Perform gradient
morph_tophat	Perform top hat
upSample	Perform up sampling by 2x
downSample	Perform down sampling by 1/2x
findFace	Detect faces(with eyes)
median	Apply median filter
lowthresh:N	Set canny low threshold to N
highthresh:N	Set canny high threshold to N
minimimCircleDis:N	Set hough circle's minimum inter circle distance to N
minimumRadius:N	Set hough circle's minimum Radius to N
maximumRadius:N	Set hough circle's maximum Radius to N
RHO:N	Set hough distance to line to N
houghPthresh:N	Set hough threshold to N
minimumLineLength:N	Set hough lines' minimum length to N
maximumLineGap:N	Set hough lines' maximum gap to N
editKernel:N	Open Kernel Wizard(N is arbitrary)
neighbourhood:N	Set neighbourhood size to N
sobelMode:N	N=1:Magnitude map N=2:Orientation Map N=3:Sobel edges
aperatureSize:N	Set aperture size

3.5.1 Known bugs

Some of the bugs are:

- not all operation work in grayscale images
- logo is not transparent right now
- some video processing can only be done through console command
- ROI when selected on a subwindow which when maximized will not reposition the ROI. It is an issue to be fixed later.
- Editing kernel for a video via console command will pop up a dialog which will not be closed even if the main window has not been closed.

Please report any bugs or issues to **manandhar.sandeep@gmail.com**



Bibliography

References

- <http://docs.opencv.org/>, OpenCV Online Documentation, last accessed at
- Robert Laganière, “OpenCV 2 Computer Vision Application Programming Cookbook”. 2011



Index

- , 17
- Brightness and Contrast, 13
- Calibration, 18
- Canny, 15
- Color Space, 13
- Console, 11
- Console commands, 26
- Contours, 15
- Crop, 14
- Custom filters, 15
- Face Detection, 21
- Filters, 15
- Fundamental Matrix, 19
- Histogram equalization, 14
- Histogram Visualization, 12
- Homography, 20
- Hough line and circle, 16
- Images and Videos, 11
- Laplacian, 15
- Miscellaneous, 12
- Pre-made filters, 15
- Settings, 17
- Sobel, 15
- SolvePnP, 19
- Up/Down sampling, 14
- User Interface, 10
- Watermarking, 13
- Workspace, 10