Assignment 2

by **Pazim Goyal**

40069412

Language Used: Python

Editor: Python IDE

Python Version: 3.6.0

OpenCV version: 3.4.5

Library used openCV2 , NUMPY

Images Used: graf/img1.ppm , graf/img2.ppm

Threshold : maximum value in grid * .4 times (code threshold=b*.4) (for wall paintaing image)

Code explanation:

Functions and Explanation:

def gradientCalculation(image): Calculate the gradient of the image (dx,dy) and calculate Ixx =dx*dx and so on for Iyy and Ixy. Apply gaussian filter to all

def keypointcal(Ixx,Iyy,Ixy,window_size,k,threshold): this functions calculate corner points and get keypoins matrix in following was:

1. Sum of window
- Using double for loop get a window around every pixel
- Get sum of that window
- Append at anchor point
- Calculate harris corner function for each trace and det calculated
2. Keep values greater then threshold
3. Non-maximum suppression applied to all the pixels
4. Get the x,y co-ordinates for all the values larger then threshold

Keypoints are marked on img 2 and img3 using opencv.drawKeypoints

Then Descriptors are calculated using following function:

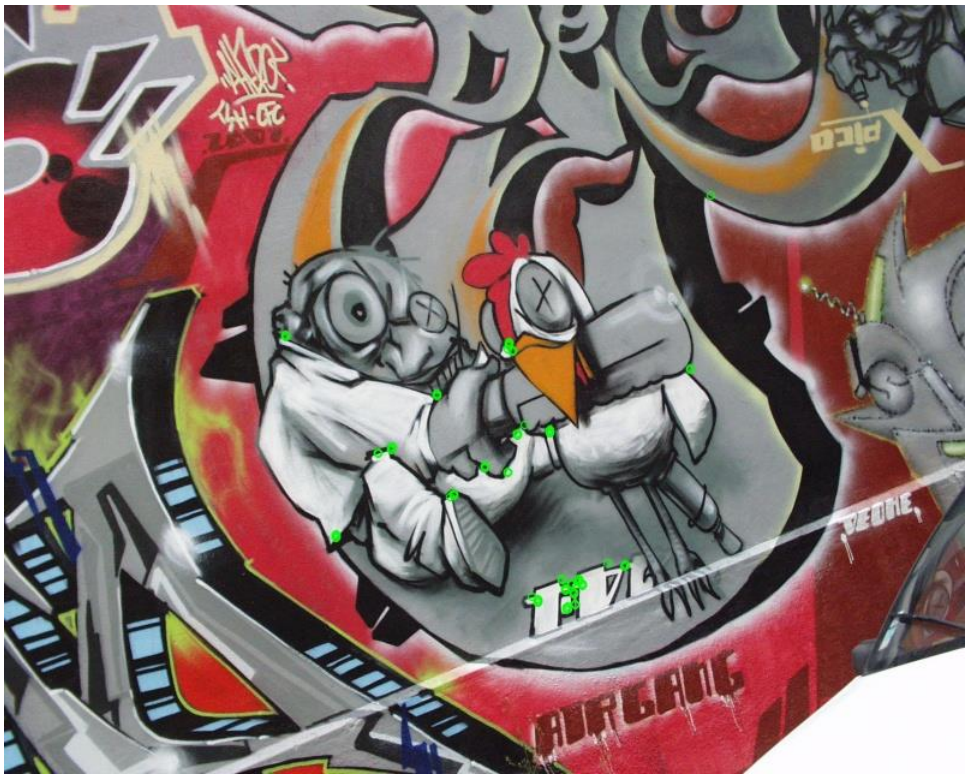
Get the magnitude and orientation for full image

- For each keypoint:
  Get the 16 X 16 window
- For every 4x4 windows in 16X16 pixels
  For every pixel get the angle (0-360), for every corresponding angle split the magnitude between some bins and for next 4x4 window append the next bin.
  At last for all the 16 windows of 4X4 window we have 128 size list.
  Normalized: this bin and clipped the values to 0.2 by adding the square of values and dividing whole bin from by root of this square.

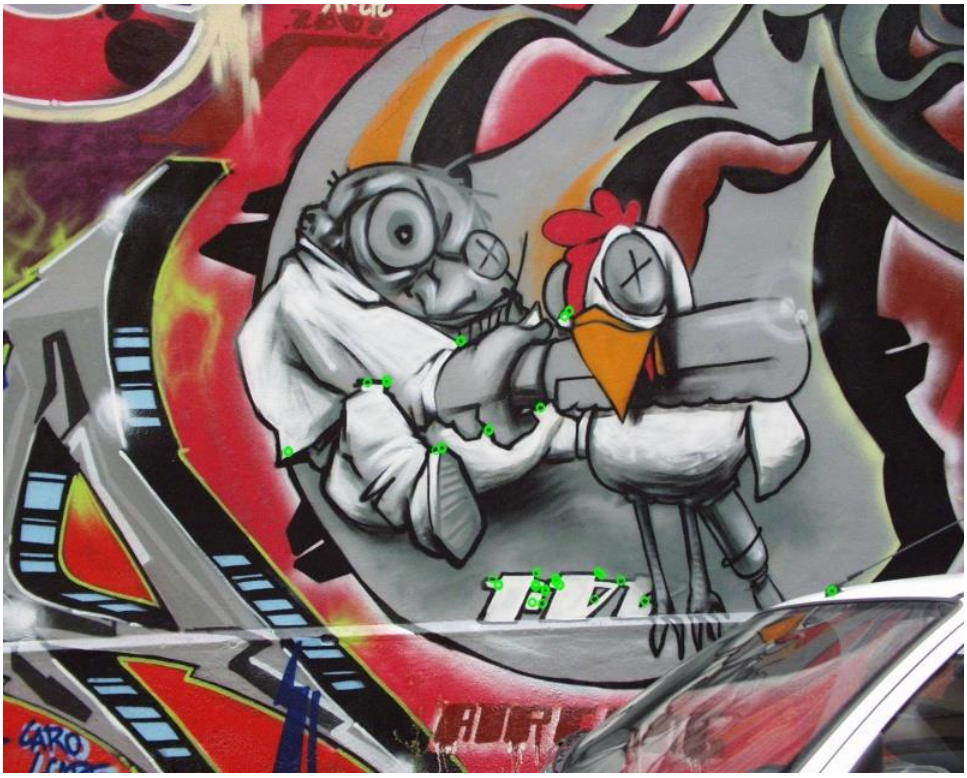
def match(binarray,binarray2): this function has two main features

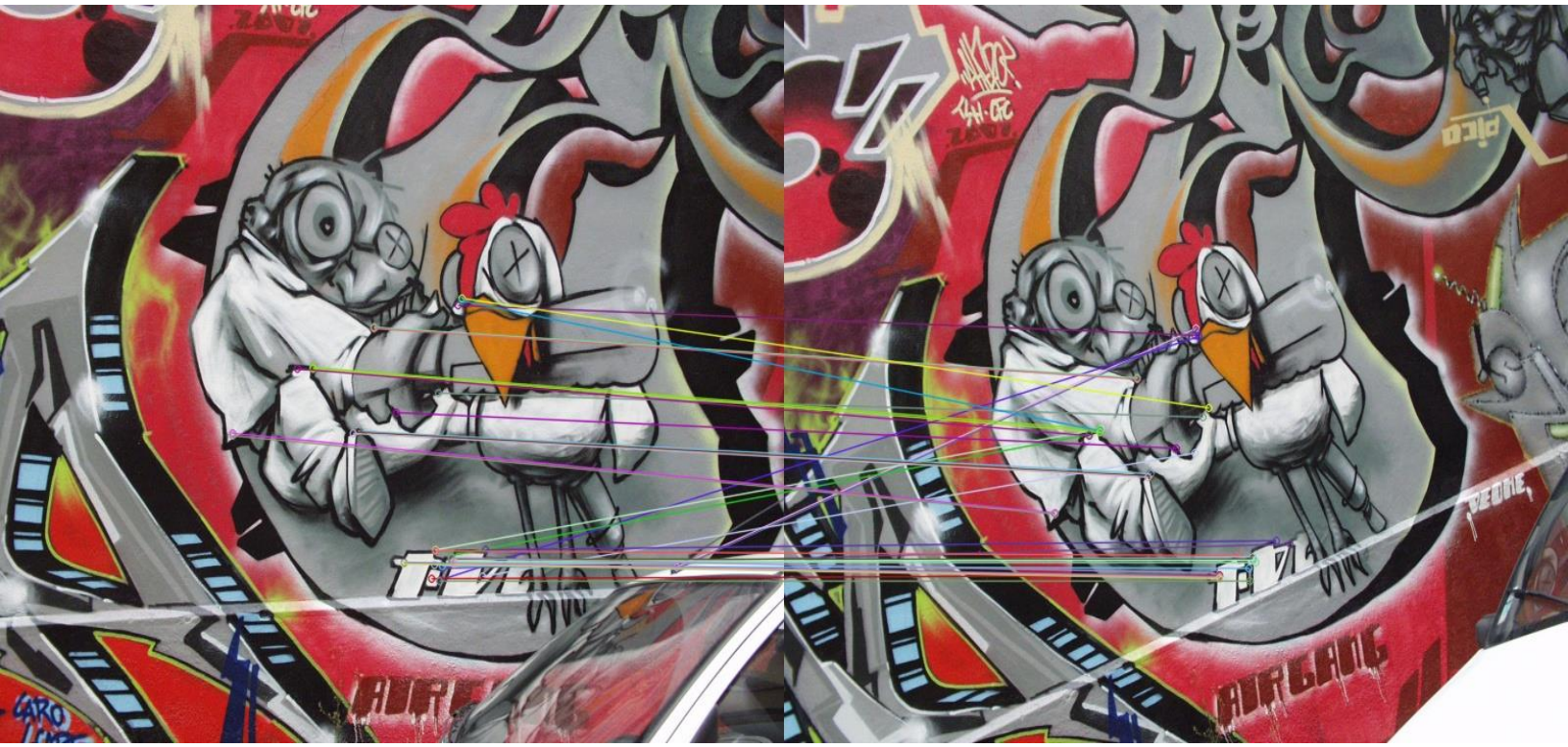
1. Calculate Sum Of Square Difference for every pair of descriptor
2. Removing bad matches of descriptors using **custom method**

Everything is saved as image1.jpg, image2.jpg , image 4.jpg in main folder



Keypoints in image 1





matches