JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

MINOR PROJECT

विद्या तत्व ज्योतिसम:

REAL TIME OBJECT TRACKING

Submitted to:	**Submitted by:**	
Ms. Ankita Wadhwa	Aditya Srivastava	14103167
Ms. Megha Rathi	Garvit Agrawal	14103115
	Vaibhav Garg	14103118

CONTENTS

ABSTRACT

The Continuously Adaptive Mean Shift Algorithm (CamShift) is an adaptation of the Mean Shift algorithm for object tracking that is intended as a step towards head and face tracking for a perceptual user interface. In this project, we review the CamShift Algorithm and extend a default implementation to allow tracking in an arbitrary number and type of feature spaces.

OBJECTIVES

Track the movement of an object by using webcam. Device a method to detect the position of the object accurately.

Several applications to run on the basis of the object tracking.

1) Game which we can be controlled by arrow keys.
2) Slide Presenter – We can run the PowerPoint presentation by any trackable object.
3) Scroller – We can scroll the browser.
4) Air paint – One can draw the imagination in the painting software.

BACKGROUND WORKS AND FINDINGS

Our project is basically based upon cam shift algorithm

First of all, we studied cam shift algorithm after we thought that this algorithm is very useful for games. Then we started implementing game.

The main task is to integrate game with cam shift algorithm

So that we can easily track object and play the game.

We decided that some more things can be included in our project then so we included these

SLIDE PRESENTER

By this we can slide show the respective PowerPoint presentation by the use of any trackable object

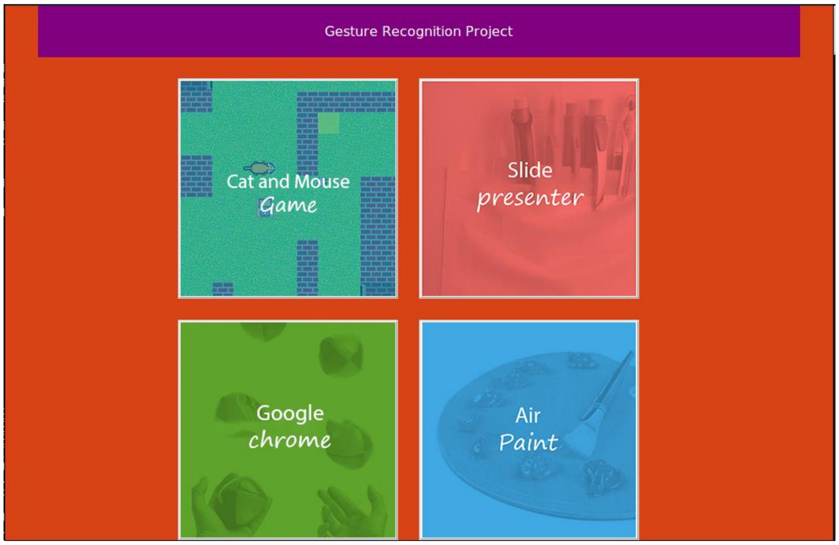
SCROLLER

By this we can scroll the web browser.

AIR PAINT

By this we can just paint by waving the trackable object in front of the webcam

DESIGN AND IMPLEMENTATION



The UI is designed using Tkinter library for python. Using the top and bottom frames we place the buttons on the frame. There are four buttons for calling different functions. We have added the respective picture of our function to the buttons for easy identification.

```
frame1 = tk.Frame(root,bg=bkgcolor)
button1 = tk.Button(frame1, width=255, height=255, image=photo1,text="Cat and
Mouse Game",fg="black",bg="white")
button1.pack(side=tk.LEFT)
button1.image = photo1
```

Upon clicking any button, function of that particular program is called.

For tracking the object, we used camshift algorithm implemented using OpenCV and Python

In this algorithm, following steps are followed:

1. Set the region of interest (ROI) of the image
2. The selected location is the target distribution to be tracked.

3. Calculate a color probability distribution of the region centered at the Mean Shift search window.
4. Iterate Mean Shift algorithm to find the centroid of the probability image. Store the zeroth moment (distribution area) and centroid location.
5. For the following frame, center the search window at the mean location found in Step 4 and set the window size to a function of the zeroth moment. Go to Step 3.

```
def selectROI(event, x, y, flags, param):

    global frame, roiPts, inputMode


    if inputMode and event == cv2.EVENT_LBUTTONDOWN and len(roiPts) < 4:
        roiPts.append((x, y))
        cv2.circle(frame, (x, y), 4, (0, 255, 0), 2)
        cv2.imshow("frame", frame)
```

After these steps, we get a rectangle box showing our tracked object and by which we can perform our respective functions.

```
camera = cv2.VideoCapture(0)

cv2.namedWindow("frame")
cv2.setMouseCallback("frame", selectROI)
```

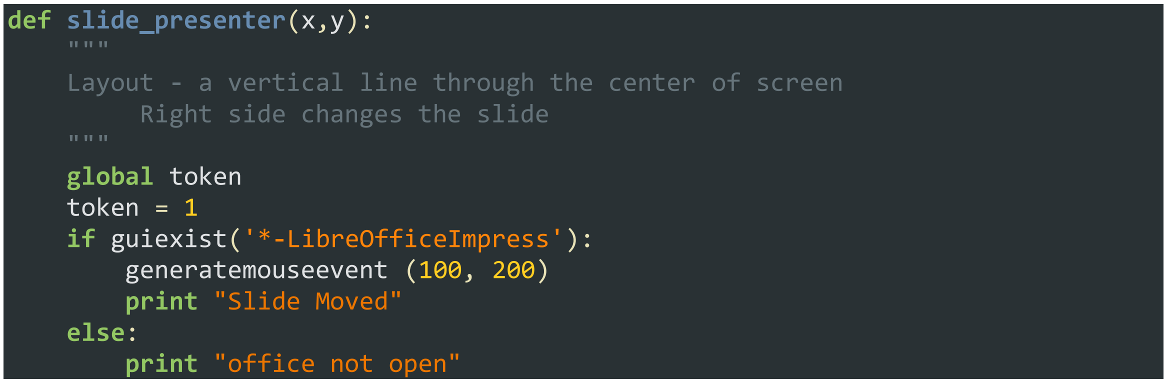
CAT AND MOUSE GAME

This game consists of a mouse which tries to collect all the cheese without getting caught by the cat. Our Object Tracking algorithm is used in the mouse to control it. The left, right, up and down movement are controlled by any the tracked object. The UI of this game is made by using various pictures related to cat and mouse in random order. There are several level in the game and for each level, the pictures are randomly placed to give a new look to the stages. The program also keep track of the scores of cat and mouse.

```
pygame.init()
screen = pygame.display.set_mode((800,640))
pygame.display.set_caption("A Game of Cat and Mouse")
```

SLIDE PRESENTER

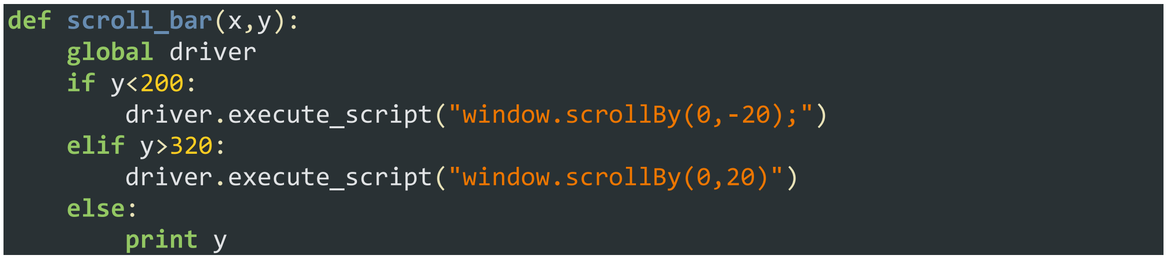
We can use this to show the respective PowerPoint presentation by the using the tracked object. Every time user waves from right to left, it clicks the left mouse button which results in moving to the next slide. We have used Ldtp for mouse click event which results in working of this presenter. In this way, one can study and change slides while lying on the bed.



```
def slide_presenter(x,y):
    """
    Layout - a vertical line through the center of screen
         Right side changes the slide
    """
    global token
    token = 1
    if guiexist('*-LibreOfficeImpress'):
        generatemouseevent (100, 200)
        print "Slide Moved"
    else:
        print "office not open"
```

SCROLLER

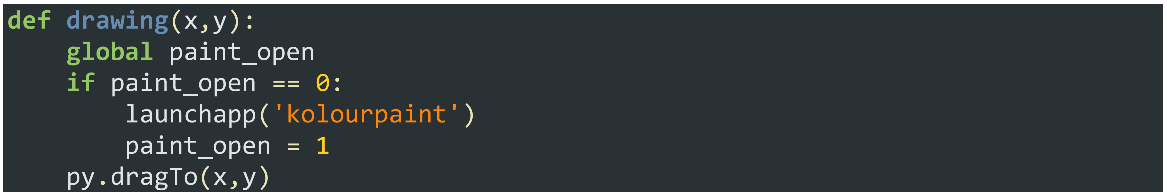
By which we can scroll the web browser. The task of opening google chrome is done with the help of Selenium. After that we run a JavaScript function scrollBy to scroll according to the movement of the tracked object. We have defined the same speed for the UP and DOWN movement. This was done by assigning (-20) for the DOWN and (+20) for the UP movement in the scrollBy function.



```
def scroll_bar(x,y):
    global driver
    if y<200:
        driver.execute_script("window.scrollBy(0,-20);")
    elif y>320:
        driver.execute_script("window.scrollBy(0,20)")
    else:
        print y
```

AIR PAINT

The name “air” comes because we can paint without touching anything on the mouse pad or screen. We can just paint by waving the tracked object in front of the webcam. The paint program launches with the ‘launchapp’ function and after that when we move the tracked object, the x and y coordinates are tracked and by using the ‘dragTo’ function we are able to paint with the tracked object which results in some beautiful drawings. Several options of paint can also be selected by just moving the object and placing it on the menu option.



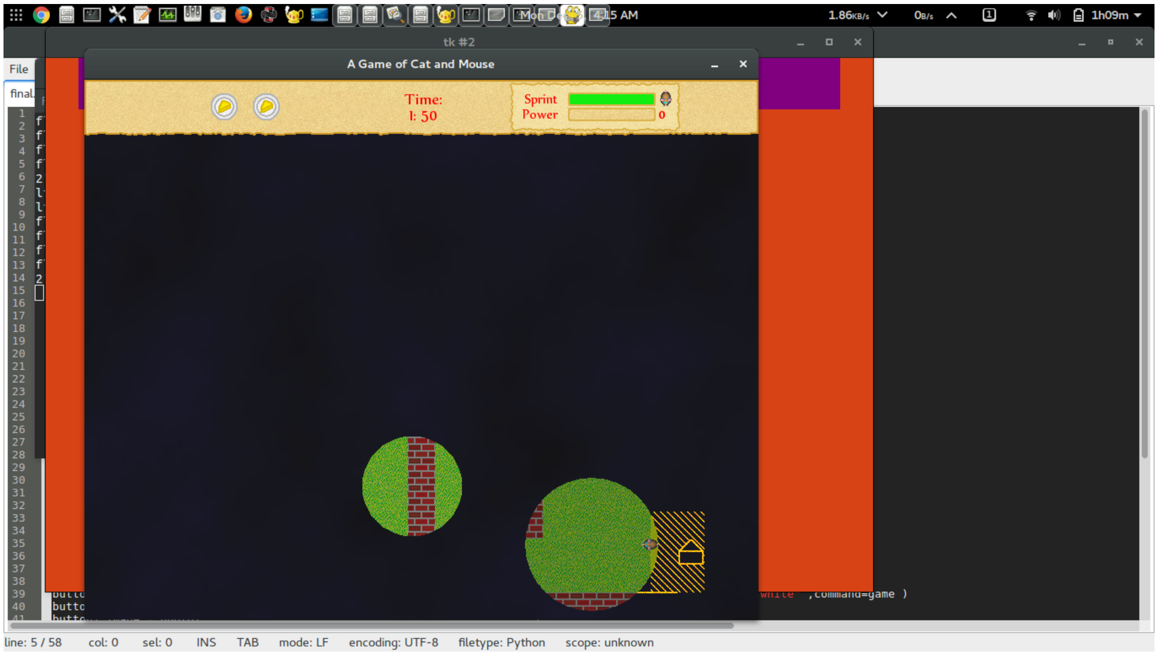
```
def drawing(x,y):
    global paint_open
    if paint_open == 0:
        launchapp('kolourpaint')
        paint_open = 1
    py.dragTo(x,y)
```

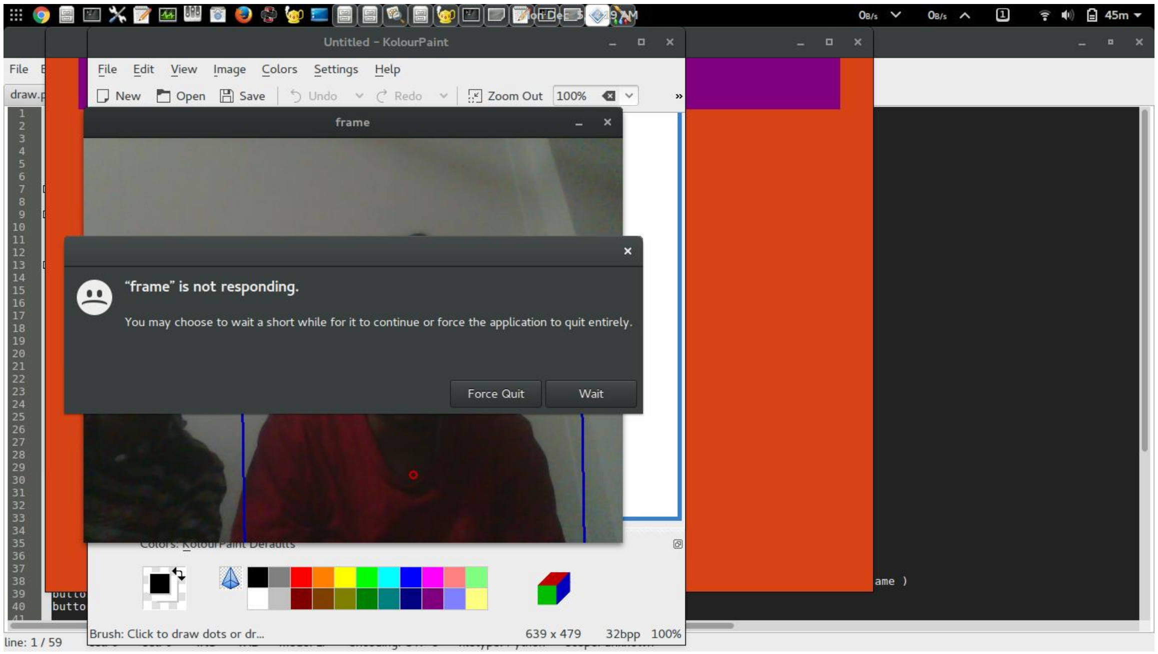
TESTING

Particulars	**Expected Output**	**Actual Output**	**Pass**	**Fail**
On click Cat and Mouse Button	1) First tracking frame should open. 2) Starts the game perfectly	1) Frame is opening and tracking the object. 2) Game is starting perfectly	1) Yes 2) Yes	
On click Slide Presenter Button	1) First tracking frame should open. 2) Display whether office is opened not. 3) Slide should change with the wave of an object.	1) Frame is opening and tracking the object. 2) Message is shown on the terminal. 3) Slide is changing	1)Yes 2)Yes 3)Yes	
On click Google Chrome Button	1) First tracking frame should open. 2) Chrome browser open.	1) Frame is opening and tracking the object. 2) Browser is opening	1)Yes 2) Yes	

On click Air Paint Button	1) First tracking frame should open. 2) Paint open.	1) Frame is opening and tracking the object. 2) Paint is opening	1) Yes 2) Yes.	
Pressing ‘q’ for quit.	1) Tracking frame should close and open main UI .	1) Window frezzes at that point.		1)Yes
Pressing ‘I’ to scan the object	1) Frame freezes for the selection of ROI points.	1) Frame freezes.	1) Yes	
>4 ROI points to select	1) Will perform perfectly.	1) Not performing		1)Yes
<4 ROI points to select	1) Does not applicable because minimum closed figure other that triangle should contain 4 points.	1) Is not applicable	1) Yes	
Press Enter to start tracking	1) Should start	1) Is starting	1)Yes	
Press any key to start tracking	1) Should start	1) Is not starting		1) Yes
On clicking the Close button in top right corner	Should Close the screen	Window is not Closing		1) Yes
Slide change	1) Next slide 2) Back slide	1)Is going on next slide 2) Is going on the next slide	1)Yes	2) Yes
Scroller	1) Up 2)Down	1)Up 2)Down	1)Yes 2)Yes	

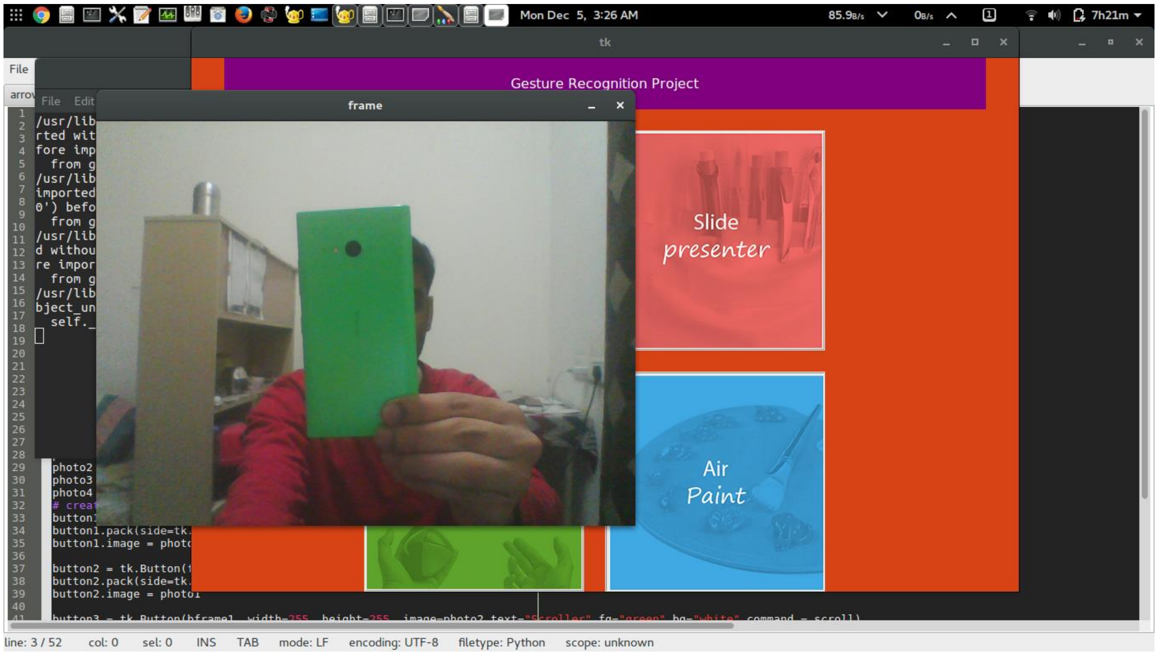
Mouse Movement	1) Movement in four directions	1) Moving in all the four directions	1) Yes	
Paint tool	1) Tool should hold on .	1)Dragging continuously instead of hold.		1)Yes





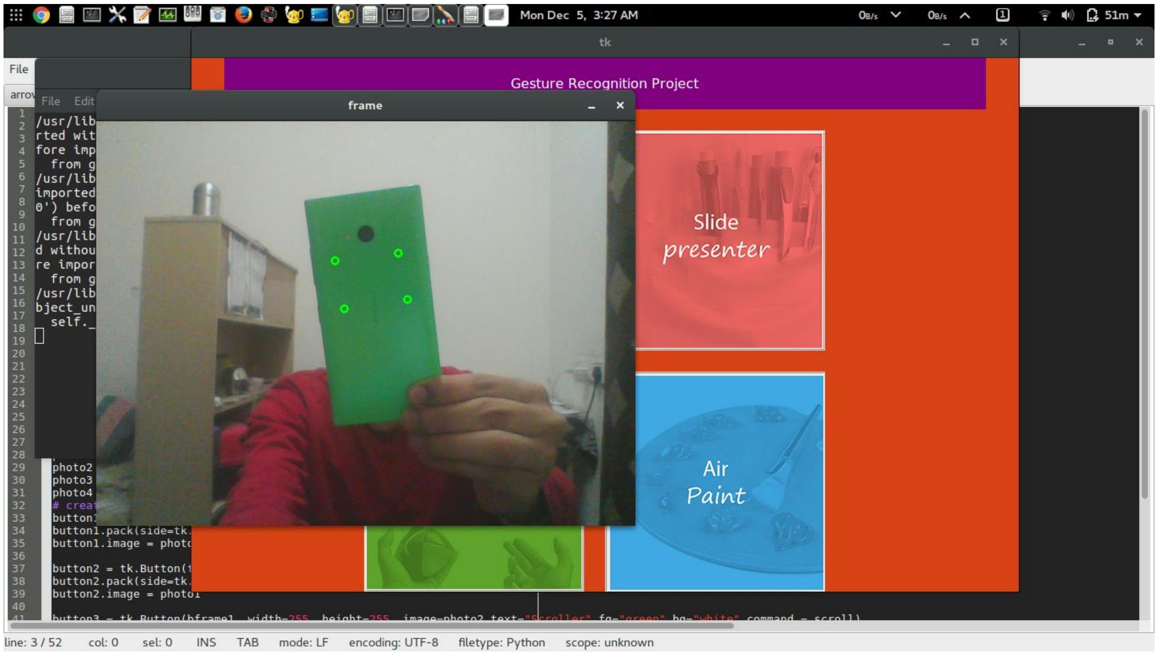
"frame" is not responding.

You may choose to wait a short while for it to continue or force the application to quit entirely.



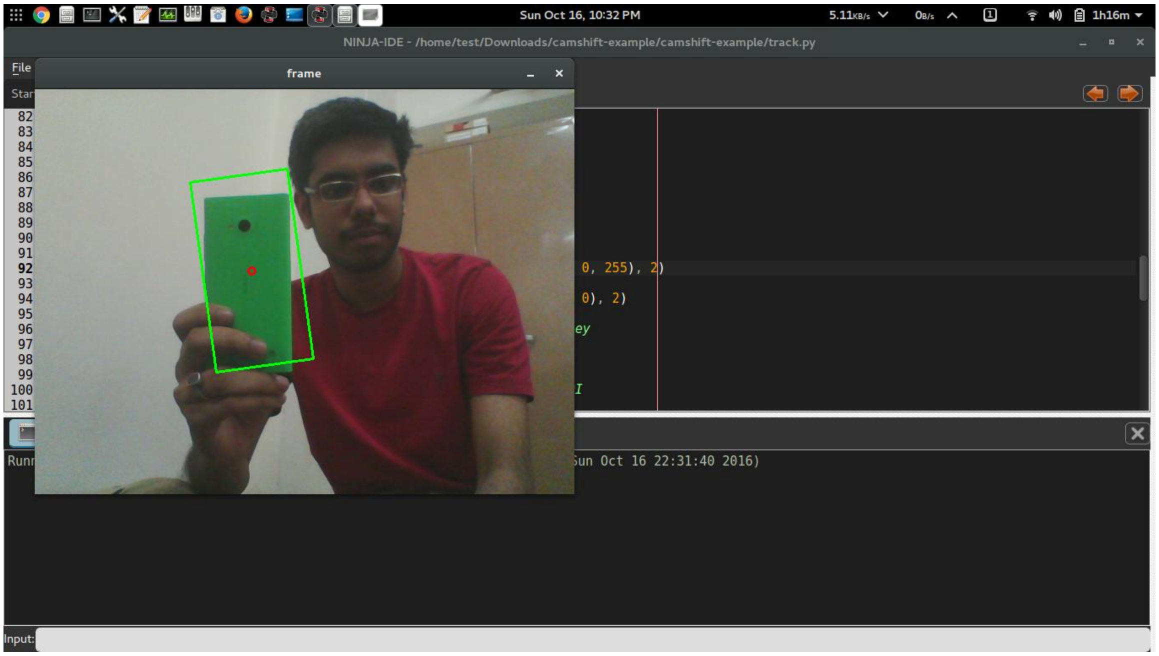
```
/usr/lib
rted wit
fore imp
  from g
/usr/lib
imported
0') befo
  from g
/usr/lib
d withou
re impor
  from g
/usr/lib
bject_un
  self._

photo2
photo3
photo4
# creat
button1
button1.pack(side=tk.
button1.image = photo

button2 = tk.Button(f
button2.pack(side=tk.
button2.image = photo1

button3 = tk.Button(bframe1, width=255, height=255, image=photo2,text="Scroller", fg="green", bg="white", command = scroll)
```

line: 3 / 52   col: 0   sel: 0   INS   TAB   mode: LF   encoding: UTF-8   filetype: Python   scope: unknown



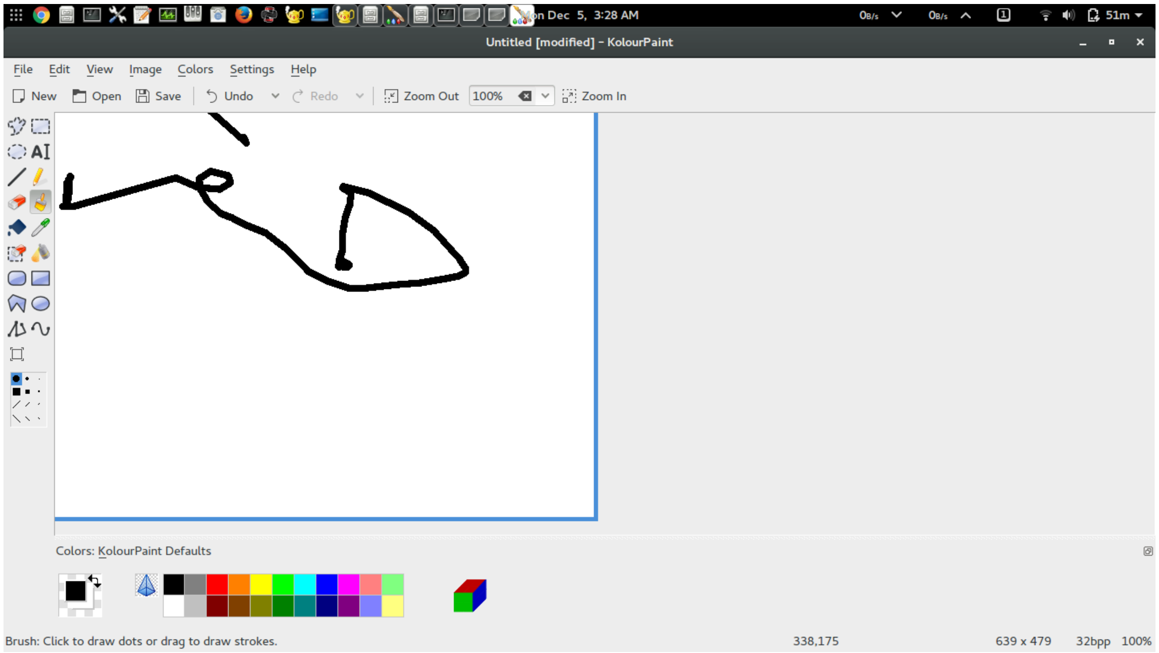
```
/usr/lib
rted wit
fore imp
  from g
/usr/lib
imported
0') befo
  from g
/usr/lib
d withou
re impor
  from g
/usr/lib
bject_un
  self._

photo2
photo3
photo4
# creat
button1
button1.pack(side=tk.
button1.image = photo

button2 = tk.Button(f
button2.pack(side=tk.
button2.image = photo1

button3 = tk.Button(bframe1, width=255, height=255, image=photo2,text="Scroller", fg="green", bg="white", command = scroll)
```

line: 3 / 52   col: 0   sel: 0   INS   TAB   mode: LF   encoding: UTF-8   filetype: Python   scope: unknown


Sun Oct 16, 10:32 PM
5.11KB/s
0B/s
1h16m
NINJA-IDE - /home/test/Downloads/camshift-example/camshift-example/track.py
File
Star
frame
Runn
Input:

```
0, 255), 2)
0), 2)
ey
I
```

```
Sun Oct 16 22:31:40 2016)
```



Brush: Click to draw dots or drag to draw strokes. 338,175 639 x 479 32bpp 100%

LIBRARIES USED

1. OpenCV
2. NumPy
3. Tkinter
4. Pyautogui

5. Ldtp
6. Selenium
7. Pygame

CONCLUSION

Camshift algorithm can be effectively used to track objects. It gives accurate positions of the objects which can be used to control other functions of a computer

FUTURE WORKS

Slide presenter-

1. One can change slide study while lying on the bed
2. Can be used to present slides in the office without the need of expensive presenters
3. Used to change slides during in classroom without the movement of teacher

Scroller-

1. One can use browser scrolling while lying on the bed or sitting apart from system

Cat and mouse-

1. Other systems that use human input are very expensive (Xbox one, ps3). Our implementation saves the end user this cost

Air paint-

1. One can paint by siting apart from the system
2. Greater control and feel for the end user upon using the system

REFRENCES

PAPERS

[1] Gevers, Theo, and Arnold WM Smeulders. "Color-based object recognition." Pattern recognition 32.3 (1999): 453-464.

[2] Allen, John G., Richard YD Xu, and Jesse S. Jin. "Object tracking using camshift algorithm and multiple quantized feature spaces." Proceedings of the Pan-Sydney area workshop on Visual information processing. Australian Computer Society, Inc., 2004.

[3] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.

INTERNET SOURCES

[1] OpenCV: Meanshift and Camshift: (http://docs.opencv.org/3.1.0/db/df8/tutorial_py_meanshift.html)

[2] Ldtp library for python: (https://ldtp.freedesktop.org/user-doc/index.html)

[3] Selenium library for python: (http://selenium-python.readthedocs.io)

[4] PyAutoGui for python: (https://pyautogui.readthedocs.io)

[5] StackOverflow for debugging: (http://stackoverflow.com/questions/40828542/opencv-window-not-closing-after-calling-destroyallwindows-function/40849011#40849011)

DIVISION OF WORK

1. Aditya
    a. Slider.py
    b. Presenter.py
    c. Draw.py
    d. Arrow.py
    e. Final code integration
2. Vaibhav
    a. Main.py – cat and mouse game code and UI
    b. Final.py – UI of the project
    c. Final code integration
3. Garvit
    a. Cat.py
    b. Mouse.py