# **DPS920/CVI620 – Lab 9**

# **Motion Detection**

| Total Mark: | 10 marks (3% of the total course grade)   * 6 out of 10: Learn@Seneca submission (Due: Wednesday November 15th end of day) * 4 out of 10: Lab demo (Due: During Workshop of week 10) |
| --- | --- |
| Submission file(s): | * Lab09\_1.py / Lab09\_1.ipynb * Lab09\_2.py / Lab09\_2.ipynb * Lab09.docx |

Please work in **groups** to complete this lab. This lab is worth 3% of the total course grade and will be evaluated through your written submission, as well as the lab demo. During the lab demo, group members are *randomly* selected to explain the submitted solution. Group members who are not present during the lab demo will lose the demo mark.

Please submit the submission file(s) through Learn@Seneca. ALL team members must submit the final work.

***Please paste the resulting images and answers in this document.***

## **Part I: Motion Detection with difference image**

1. Write a code in Lab09\_1.py to use the webcam video stream (similar to Lab 1). Apply the following changes:
2. Convert the captured frames to grayscale.
3. Save the first frame (converted to grayscale) as background image.
4. Change code to display the difference images, instead of captured frames.
5. Threshold the difference image using a threshold of 128.
6. Count the number of pixels changed by at least 128 gray levels.

Hint: You can do this by finding the sum of the threshold image in part d.

1. If the count is larger than 1.5% of the number of pixels in the frame,
   1. Output the time the activity was detected.
   2. Save the current color frame in a file (with a counter or time stamp).
   3. Wait for 5 seconds before returning to capturing loop.
   4. Update the background image.
2. Exit the program by entering esc or ‘q’.
3. Adjust the motion sensitivity to your desired sensitivity, e.g., 1%, 2%, … of the frame size.
4. Paste three samples of background and the difference image here:
   1. **one with the default sensitivity, e.g., 1.5%**

A close-up of a couch

Description automatically generated

A close-up of a hand

Description automatically generated

* 1. two with your desired sensitivity.

**Sensitivity = 2:**

A close-up of a person's back

Description automatically generated

A close-up of a ring on a finger

Description automatically generated

**Sensitivity = 3:**

## A close-up of a hand Description automatically generated

A close-up of a glove

Description automatically generated

## **Part II: Motion Detection Using Optical Flow**

1. Copy code in Lab09\_1.py to Lab09\_2.py. Then change the code to use the magnitude of the dense Farneback optical flow to detect motion.
2. Visualize the optical flow vectors and paste samples here. Show the results using the color-coded results and the vector of object displacement.

A computer screen shot of a person

Description automatically generated

A colorful lines on a black background

Description automatically generated

1. Explain what is being done in this part and how it differs from Part I.

In Part I, motion is detected by computing the absolute difference between consecutive frames and applying a threshold. The sensitivity parameter determines the percentage of changed pixels to consider as motion.

In Part II, Farneback optical flow is used to calculate dense motion vectors between consecutive frames. The magnitude of these vectors is used to create a motion mask, highlighting regions with significant motion.

We continuously calculate the optical flow using the Farneback method, then convert the optical flow vectors into polar coordinates to get magnitude and angle. Then, we use HSV color space to represent the optical flow vectors. Hue represents the direction (angle), saturation is constant, and brightness represents the magnitude. Displacement vectors are represented as arrows on the visualized frame.

Once significant motion is detected (using magnitude threshold), the image is being saved.

Hint:

# Convert optical flow into Polar coordinates to get magnitude

mag, ang = cv.cartToPolar(flow[..., 0], flow[..., 1])

# Use a threshold, to only count the significant ones

mag\_thresholded= (mag > 20)

percent = mag\_thresholded.sum() / (frame\_width \* frame\_height)

## **Part III: Group Work**

1. Add this declaration to your file:

We, group 5, Davender Singh and Liliya Panfilova, declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. We have not copied any part of this assignment, manually or electronically, from any other source including web sites, unless specified as references. We have not distributed our work to other students.

1. Specify what each member has done towards the completion of this work:

|  | Name | Task(s) |
| --- | --- | --- |
| 1 | Davender Singh | Part 1 |
| 2 | Liliya Panfilova | Part 2 |
| 3 |  |  |