# **CVI620/DPS920 - Lab 4**

# **Transformations and Filtering**

| Total Mark: | 10 marks (3% of the total course grade)   * 6 out of 10: Learn@Seneca submission (Due: Wednesday February 8th end of day) * 4 out of 10: Lab demo (Due: During Workshop of week 5) |
| --- | --- |
| Submission file(s): | * Lab04\_1.py / Lab04\_1.ipynb * Lab04\_2.py / Lab04\_2.ipynb * Lab04.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 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: Geometric Transformations**

1. Write code (save it as Lab04\_1) such that:
   1. Open an image (paste a sample here).
   2. In a loop, asks the user whether s/he wants to rotate, resize, apply perspective transformation, or exit.
   3. If **rotation** is selected, asks for an angle (in degrees). Then rotates the image around the center of the image with the given angle and displays the rotated image. Run the code with
      1. Rotate for 20 degrees clockwise (paste the sample here).
      2. Rotate for 10 degrees counter clockwise (paste the sample here).
   4. If **resizing** is selected, asks for the resizing factors along each axis and resizes the image. Run the code resizing the image to double the width and half the height (paste the sample here).
   5. If **perspective** **transformation** is selected, apply the transform with the following homography matrix and use same image size for output image (paste the sample here)

## **Part II: Filtering**

1. Create a program (save it as Lab04\_2). Include code to:
   1. Open an image (paste a sample here).
   2. Ask for a probability value between 0 and 1 and implement the salt and pepper noise with the given probability. Create a noisy image with p=0.1 (paste the sample noisy image here).

Hints:

* The noisy image is the same as image except at affected pixels.
* The number of pixels affected by noise is p \* width \* height of image.
* The location of each affected pixel (row and column) is random within possible row and column values.
* The color of the noise is a random gray color.
  1. Smooth the above noisy image using a 3 x 3 box kernel (paste the sampled filtered image here).
* If you like, choose a pixel and output the pixel’s blue value, as well as its neighbors’. Calculate what you expect for the blurred pixel value. Then check the value for the blurred image. Is it what you expect?
  1. Filter the above noisy image using a 3 x 3 bilinear filter (paste the sampled filtered image here).
  2. De-noise the noisy image using a 3 x 3 median filter (paste the sampled filtered image here).
  3. Compare the results of the above filters. Which filter works better for de-noising? Why is that?
  4. De-noise the noisy image using a 3x3 Gaussian filter with sigma = 1.5 (paste the sampled filtered image here).
  5. Try different kernel sizes- for example (5x5), (7x7)- and different sigma values – for example, sigma = 0.5, 1.0, 3.0, 5.0- for the Gaussian filter. How do the results change? (No need to paste results for this part)

## **Part III: Group Work**

1. Add this declaration to your file:

We------------ (mention assigned group number and your names), 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 |  |  |
| 2 |  |  |
| 3 |  |  |