# **CVI620/DPS920- Lab 2**

# **Creating images and capturing video**

| Total Mark: | 10 marks (2.5% of the total course grade)   * 7 out of 10: Blackboard submission (Due: End of week 2) * 3 out of 10: Lab demo (Due: During lab of week 3) |
| --- | --- |
| Submission file(s): | * Lab02.cpp * Lab02\_20W.docx |

Please work in **groups** to complete this lab. This lab is worth 2.5% 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 Blackboard. Only one person must submit for the group.

## **Part I: A photo booth application**

For this lab, you need a webcam or a digital camera connected and installed on your machine.

1. Start a New Project in VS ( C++, Console Application) and name it Lab02.
2. Copy and Paste the property sheet file created in Lab 1 (PropertySheet\_debug.props) to the project folder (same folder as Lab02.vcxproj file). Then follow the following steps to add them to the project.
   1. Go to View > Other windows > Property Manager
   2. Right click on Lab02 > Debug| x64 and choose Add Existing Property Sheet
   3. Select PropertySheet\_debug.props file and click Open.
3. Replace the code in Lab02.cpp with the code from example\_02-10.cpp at <https://github.com/oreillymedia/Learning-OpenCV-3_examples>.
4. Choose configurations of Debug and x64. Then build and run the project. You should see your webcam feed on your screen.
5. Change the code to:
   1. Use std::cout to output the width and height of the video capture object using GET member function.
   2. When the ‘x’ key is pressed, save a snapshot. Use image names such as ‘image01,jpg’, ‘image02.jpg’, etc., automatically incrementing the filename counter. While saving, freeze the image for 1 second.
   3. When the ‘Enter/Return’ key or the space key are pressed, exit the application.
   4. Ignore all other keys.

## **Part II: Basic calculations**

1. Calculating fps:
   1. Calculate the fps implemented in the code using waitKey (ignore all other delays including 1 second freezes).

1000/33 = 30.30

It is 30 fps

* 1. How would you change the code to have an fps of 25 frames per second?

1000/x = 25

1000/40 = 25

X = 40

Change the waitKey to 40 ms

1. Calculating Compression Ratio:
   1. Find image01.jpg and look at its properties:
      * What is the image resolution (dimensions)? Does this match the width and height output of question 5?
        + 640 x 480 which is the same as what was printed in output.
      * What is the bit depth?
        + 24
      * What is the file size in bytes?
        + 79,273 bytes
      * If this image was not compressed, what would the file size be?
        + 640 pixels x 480 pixels x 3 bytes = 921, 600 bytes
      * Calculate the compression ratio as the ratio between the uncompressed size and the compressed size.
        + 921,600:79,273
        + 921,600 / 79,273
        + 11.6:1
        + Compression ratio of 11.6
2. Add this declaration to your file:

We, Jason, Muqing, and Matteo, 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 | Jason | Code and Calculation |
| 2 | Muqing | Code and Calculation |
| 3 | Matteo | Code and Calculation |