# **CVI620/DPS920- Lab 9**

# **Segmentation**

| Total Mark: | 10 marks (2.5% of the total course grade)   * 7 out of 10: Blackboard submission * 3 out of 10: Lab demo |
| --- | --- |
| Submission file(s): | * Lab08.docx including the **result images and answers** |

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 ***(include all results)***, 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.

1. Add this declaration to your file:

We, ------------ (mention 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.

Start a New Project in VS12: Visual C++ > Win32 Console Application > Empty project and name it Lab09. Add the debug and release property sheets as in Lab 1.

## **Part I: The watershed algorithm**

1. Use the sample code for the watershed algorithm: <https://docs.opencv.org/3.0.0/d8/da9/watershed_8cpp-example.html>

Run and test the code with the given sample image. Using the mouse, mark the segments (see example on slide #22). How do your markers guide the algorithm? How is the performance of this algorithm? Paste the segmentation results here.

## **Part II: The Grab-cut algorithm**

1. Use the sample code for the grab-cut algorithm: <https://docs.opencv.org/3.0.0/de/dd0/grabcut_8cpp-example.html>

Run and test the code with the given sample image. Compare the performance of this algorithm with the method used in part I. Paste the segmentation results here.

## **Part III: The mean-shift algorithm**

1. Use the sample code for the mean-shift algorithm:

<https://raw.githubusercontent.com/kipr/opencv/master/samples/cpp/meanshift_segmentation.cpp>

Run and test the code with the given sample image. Compare the performance of this algorithm with the methods used in part I and II. Which values for the parameters result in a better segmentation? Paste the segmentation results here.

## **Part IV: K-means**

1. Use K-means to segment the sample image. Use the code in :

<http://answers.opencv.org/question/27808/how-can-you-use-k-means-clustering-to-posterize-an-image-using-c/>

Experiment with 2, 3, 5, and 7 clusters. Paste the segmentation results here.

What are the limitations of k-means in this image? What are the advantages? Are the initial centers selected randomly? What is the ‘attempts’ parameter and why is it important?

Can you suggest some image processing on the image that could improve the k-means segmentation results? Apply your suggestion and paste results here.

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

|  | Name | Task(s) |
| --- | --- | --- |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |