Course: 282.762 Robotics and Automation

Assessment: Assessment 4: Written Assignment

Course Learning Outcomes Assessed:

- Use artificial vision and optimization methods in artificial vision.

Weighting: 30 %

Due Date: 17/05/2019

This is an individual assessment.

Dr Frazer K. Noble
Massey University
Auckland

Introduction

Machine vision allows for systems to make decisions based on visual information, e.g. colour, shape, size, and orientation. In this assessment, you are required to compute the locations of an object's instances in a robot's workspace, compute the shortest path between all the instances, and display the results in a Graphical User Interface (GUI).

Aims

The assessment's aims are to:

- Learn how to use the Open Computer Vision (OpenCV) Library to localise objects in an image.
- Practice using the Qt Framework (Qt) to develop a GUI and display an image.
- Practice developing your own algorithms for solving unique problems.

Objectives

The assessment's objectives are to:

- Design and implement an algorithm that computes the locations of an object's instances.
- Use the algorithm to detect the location of an object's instances in a reference image.
- Design and implement an algorithm that computes the shortest distance to travel between each object, starting from point (0, 0).
- Design and implement a GUI.
- Display the object's instances' locations in the GUI.
- Display the shortest path in the GUI.
- Save the object's instances' locations and the shortest path as output.png in the project's data sub-directory.

Requirements

You are required to:

- Write a program that achieves the assessment's objectives.
- Write a report detailing what you did, how, and why.

Resources

You will be provided with the following:

- A project template.
- A reference image.

Tools

You should use the following tools to complete this assessment:

- Qt 5.12
- OpenCV 4.0.1
- CMake 3.13.4

Either:

- Visual Studio IDE
- Visual Studio Code

Dr Frazer K. Noble
Massey University
Auckland

Submission Instructions

Add all your work to a .zip archive and name it in the following format: FIRSTNAME_LASTNAME_ID.zip. Be sure to include your program's source code.

Upload your submission to Stream before the due date. A 5 % per day penalty will be applied to late submissions.

Frequently Asked Questions

- Do I need to use the software versions listed?

No, you can use older versions; however, you may find that some source code is different, e.g. OpenCV's ENUMs changed from version 3.0 to 4.0, and older versions of CMake won't find recent versions of OpenCV, etc.

- Do I need to use C++ to write my program?

Yes. We will look at support Python and JS in the future.

- Do I have to use CMake?

No, CMake just allows you to work across Windows and Linux.

- Do I have to use Visual Studio IDE or Visual Studio Code?

No, you can use whatever IDE you want; however, Visual Studio IDE's IntelliSense functionality makes coding a lot easier. Also, Visual Studio Code works on Windows and Linux and is a light-weight and flexible IDE.

- How should I display the object's instances' locations and the shortest path in the GUI?

You can draw a box around each of the object's instances over the reference image. You can display the shortest path as a series of lines drawn from the instances' centre points. You can display each instance's attributes in a text box. You can display the shortest path's value in a text box.

- How will you measure how well my program works?

I have computed the object's instances centre points and the shortest path; I will compare the length of your shortest path against my reference. Also, I will be looking at how long your application takes to load, how much memory it uses, and whether it will run without throwing any exceptions.

- Do I need to comment my source code?

No, but you will need to explain what each block of code does, how it works, and why you used that source code in your report. Commenting your code; however, will certainly make writing the report easier.

- Can I get an extension?

Yes, but only if it's for a good reason. Extensions will be granted at course coordinator's discretion.

If you have any questions, please post them on the course's Stream site.

Dr Frazer K. Noble
Massey University
Auckland

Marking Rubric

	E Range (0 – 39.99)	**D Range (40 – 49.99)**	**C Range (50 – 64.99)**	**B Range (65 – 79.99)**	**A Range (80 – 100)**	**Weighting**
	Inadequate	*Poor*	*Adequate*	*Good*	*Excellent.*	
Program	The program performs at an inadequate level. The program may only load the reference image. The GUI displays the reference image.	The program performs at a poor level. The program can load the reference image, execute part of the implemented algorithm, but may throw several exceptions. Major editing of source code may be required. The program's output image illustrates it localised the object's instances to a poor level. The GUI displays the reference image.	The program performs at an adequate level. The program can load the reference image, execute the implemented algorithm, compute the shortest path within +- 15 % of the actual value, and generate the output image, but may throw an exception. Minor editing of source code may be required. The program's output image illustrates it localised the object's instances and computed the shortest path to an adequate level. The GUI displays the reference image and	The program performs at a good level. The program can load the reference image, execute the implemented algorithm, compute the shortest path within +- 10% of the actual value, and generate the output image without throwing any exceptions. The program's output image illustrates it localised the object's instances and computed the shortest path to a good level. The GUI displays the reference image, object's instances' locations, their	The program performs at an excellent level. The program can load the reference image, execute the implemented algorithm, compute the shortest path within +- 5 % of the actual value, and generate the output image, quickly and efficiently, and without throwing any exceptions. The program's output image illustrates it localised the object's instances and computed the shortest path to an excellent level. The GUI displays the reference image, object's instances'	60%

Dr Frazer K. Noble
Massey University
Auckland

			object's instances' locations.	attributes, the shortest path, and its value.	locations, their attributes, the shortest path, and its value clearly.	

	E Range (0 – 39.99)	**D Range (40 – 49.99)**	**C Range (50 – 64.99)**	**B Range (65 – 79.99)**	**A Range (80 – 100)**	**Weighting**
	Inadequate	*Poor*	*Adequate*	*Good*	*Excellent.*	
Report	The report is written to an inadequate level. The report describes what was done.	The report is written to a poor level. The report describes what was done. The report discusses the program's output.	The report is written to an adequate level. The report describes what was done. The implemented algorithm's source code is documented, with each line's or block's purpose commented on. The report discusses the program's output.	The report is written to a good level. The report describes what was done and how. The implemented algorithm's source code is documented, with each line's or block's purpose explained. The report discusses and evaluates the program's output.	The report is written to an excellent level. The report describes what was done, how, and why. The implemented algorithm's source code is documented, with each line's or block's purpose explained in detail. The report discusses and evaluates the program's output in detail.	40%

Dr Frazer K. Noble
Massey University
Auckland