RMIT UNIVERSITY

School of Engineering

EEET2169 – Image Processing/EEET1255 – Image Systems Engineering

Laboratory Group Project

1 Aims

(i) To design, implement and test a general solution to a specific image processing problem.

(ii) To research, review, interpret and evaluate published literature on advanced image processing techniques.

(iii) To demonstrate proficiency in the practical specification, implementation, optimisation and verification of image processing operations in Matlab.

(iv) To demonstrate clear technical documentation and communication of a complete solution to a specific image processing problem.

2 Introduction

You are expected to work with one other student (groups of two, max). ***Each group must submit a combined technical report documenting their chosen image processing problem, the proposed solution/implementation and summarising their results. The report must be accompanied by a single `.zip` archive containing their implementation/source code and any associated project files.*** This laboratory project will run for three weeks, with submission due at the end of week 12 (electronic submission via the subject Canvas website; 1 week after the final laboratory session).

The aim of this laboratory project is to design and implement a general solution to a specific image processing problem or application. You will use Matlab to implement and evaluate your design using appropriate test cases. You will document your design and its verification and testing in a technical laboratory report.

Matlab and the Image Processing Toolbox [1] may be downloaded for installation on your home computer under the RMIT Total Academic Headcount (TAH) license (see [2]).

3 Requirements and Guidelines

3.1 Application/Image processing problem

You have considerable scope in this laboratory project to study an image processing problem or application of your choice. Your chosen application must allow for demonstration of knowledge and understanding of fundamental image processing concepts, as well as demonstration of competency in the design, implementation, testing and verification of image processing operations of moderate complexity. As a guide, a short list of example projects is given in Appendix A.

Projects will be weighted according to complexity/difficulty, at the discretion of the Course Coordinator.

3.2 Design considerations

In developing your solution, you should review previous published work relevant to your chosen image processing problem or application. It is expected that, if necessary, you will research and

evaluate advanced image processing operations and techniques that extend the fundamental image processing operations covered in the lecture material.

Alternative solutions should be considered where appropriate, and your design decisions clearly described. You should consider, and discuss in your report, any simplifying assumptions or limitations of your proposed solution.

3.3 Implementation

A software implementation of your proposed solution/design is required. The implementation must be a general solution to the chosen problem/application and must be tested and evaluated using a sufficiently rich and diverse set of inputs. You should consider, and discuss in your report, any limitations of your solution and your implementation of that solution.

4 Assessment process

Projects will be assessed via a combination the technical laboratory report (70%) and your software implementation (source code and documentation; 30%). The technical laboratory report must be submitted as a single .pdf file, separate to your project implementation files (submitted as a separate `.zip` archive). Submissions must be made via the subject Canvas website.

4.1 Technical laboratory report (70%)

The technical laboratory report must include a clear and concise statement of your chosen image processing problem or application. The report must then document the requirements of your chosen application, and the design, implementation and verification of your solution. The report must be complete and self-contained, and provide sufficient technical detail and supporting explanation for your solution to be understood and evaluated by an engineer without reference to external material (a guide is given in Appendix B). Limitations of your proposed solution or your implementation must be discussed.

There is no set page limit for the report. Your report should only be as long as is necessary to describe your work, and no longer. The project is a group effort but the contribution of each student (to the design and implementation, and the report) must be clearly indicated (e.g., in the report).

4.2 Implementation (30%)

Each group must submit their project implementation, i.e., source code and supporting test cases and example images, for assessment. Source code must be organized and appropriately documented (e.g., through code comments), describing input/output arguments and any usage information for each component or function.

4.3 Weighting for project complexity/difficulty

Each project will be marked out of 100 and this mark then weighted for complexity/difficulty, at the discretion of the Course Coordinator. You are highly encouraged to discuss your project and it's complexity/difficulty with the Course Coordinator. The complexity weighting, $w \in [1,5]$, (1 = low complexity, 5 = high complexity) will be used to scale the mark, k (out of 100), to a weighted mark, k^*, according to the following relationship:

$$k^* = \left\{0.9 + 0.1 \cdot \frac{(w-1)}{4}\right\} \cdot k. \qquad (1)$$

The weighted mark, k^*, will then be converted to an appropriate final percentage.

5 Late penalty

The technical laboratory report and accompanying `.zip` archive must be submitted via Canvas by 11:59pm on the due date. Late submissions will incur a penalty of 10% per day up to a maximum penalty of 50% (5 days). For example, a submission scoring a weighted mark, k^*, of 90% if submitted by 11:59pm on the due date will instead score 80% if submitted after 11:59pm on the due date and before 11:59pm the following day.

6 References

[1] Mathworks Inc., https://au.mathworks.com/products/matlab.html [online] (Accessed 2021-02-01).
[2] Mathworks Inc., The Image Processing Toolbox. https://au.mathworks.com/help/images/index.html [online] (Accessed 2021-03-07).
[3] https://rmititsm.service-now.com/kb_view.do?sysparm_article=KB0011275 [online] (Accessed 2021-03-08).

7 Appendix A

Example projects/applications:

- Change counter/totalizer [Complexity: low]. Using photographs of a collection of coins (pocket change) identify coins of each denomination and report the sum total of the coins photographed.
- Forensic document recovery [Complexity: medium]. Using scanned images or photographs of fragments of shredded documents containing images and text, segment, align and reassemble the images to recover the original documents.
- QR code detection in smartphone images [Complexity: high]. Quick Response (QR) codes are ubiquitous in cities around the world today; scanned by users with smartphones for contact tracing purposes, or to access local information. Using smartphone photographs, automatically identify QR codes for decoding.
- Your own project [Complexity: tbd]. You are highly encouraged to specify your own project or suggest a variation on one of the projects described above.

8 Appendix B

As a guide, the technical laboratory report should contain the following:

a) A title page, stating the course code and course name, the project title, and the names and student IDs of the group members.

b) Table of contents: listing section number, heading and the **page number** of each section in your report and any appendices.

c) Introduction: providing a brief overview, statement of the problem/application, and defining the scope of the project.

d) Background: containing a concise presentation of any technical material relevant to the project or its implementation. It is expected that you will briefly review any previous published work relevant to your chosen image processing problem or application

e) Methods: containing a description of the proposed solution and the software implementation. This should include a breakdown of input information, image processing operations, software components/functions, specifications/requirements for each component, as well as tables, diagrams or figures (as necessary) to document and communicate your solution and your implementation.

f) Results: presenting the outcome of testing the software components/functions, demonstrating their operation and assessing this against the requirements previously identified (see Methods).

g) Discussion and Conclusion: presenting a clear statement of the contribution of each group member to the project, as well as a discussion of the results. This should include discussion of any problems encountered and/or revisions made in arriving at the final design solution or its implementation. Limitations of your solution or its implementation must be discussed.

h) References: clearly listing complete citation information for all referenced material. You must acknowledge all sources of information used in your report. You may choose whichever referencing system you prefer, but the system must be one of those in regular use in technical journals. If in doubt, use the IEEE citation style. This list should not include general references not specifically cited in the report.

i) Appendices: containing any secondary information not essential to understanding the report. Examples might include additional test cases and outputs, program listings, derivation of novel or less well known mathematical relationships or theorems used in the report. Note that the full source code for your solution should *not* be included in the appendices – source code must be documented (e.g., using comments) and submitted electronically.

All figures and tables must be numbered and include a brief caption clearly summarising the content of the table or figure. All axes must be labelled.