

MPHY0030: Programming Foundations for Medical Image Analysis

Assessed Coursework 2 (a project) 2022-23

Available on 12th December 2020

Submission before 16:00 – 30th January 2022, on Moodle

Introduction

This is the second of the two assessed coursework. This coursework accounts for 50% of the module. Code needs to be submitted for completing a project, described below, together with a written report, describing the relevant background, methods, experiment and results.

Similar to Coursework 1 but with a focus on real-world problem solving, the aim of the coursework is to develop and assess your ability *a)* to understand the technical and scientific concepts behind the medical image analysis methods, *b)* to research the relevant methodology and implementation details of the topic, and *c)* to develop the numerical algorithms in Python and relevant libraries and packages.

The project does not specify any argument, data structure, functions, classes, modules, files or how they are designed and implemented. They will be assessed for correctness but not for their designs. Although the assessment does not place emphasis on coding skills and advanced software development techniques, basic programming knowledge will be taken into account, such as the correct use of NumPy arrays, sufficient commenting and consistent code format. Up to [20%] of the relevant marks may be deducted for good programming practice.

Do NOT use this document for any other purposes or share with others. The coursework remains UCL property as teaching materials. You may be risking breaching intellectual property regulations if you publish the details of the coursework or distribute this further.

Python packages

You can choose up to two pip-installable external packages of your choice for this coursework. No other packages should be used outside of the conda environment below.

```
conda create -n mphy0030-cw python=3.9 pillow=9.2 matplotlib=3.6 numpy=1.23 scipy=1.9 scikit-image=0.19
```

If you do use any external packages, please clearly indicate in a separate “requirement.txt” file, with pip installation commands you used in the above conda environment “mphy0030-cw”. For example:

```
pip install sklearn simpleitk
```

Up to [100%] of the relevant marks may be deducted for using external packages without being clearly specified. This will be tested on the markers’ computers with the same conda environment.

Working directory and main script

A Python main script should be a file named as “main.py”. This script can be executed within the conda environment described above, when the folder “cw2” containing the main script is used as the current working directory:

```
python main.py
```

It is the individual's responsibilities to make sure the submitted main script can be run, in the specified conda environment. Even for the data/code available in module tutorials, copies or otherwise automated links need to be provided to ensure a standalone executability of the submitted code. Up to [100%] of the relevant marks may be deducted if no runnable main script is found. N.B. The minimum that needs to be submitted is the runnable "main.py", although such a single-file script is not recommended.

The Code

The code needs to be able to re-produce what is included in the report, described below.

The Report

The report should not exceed 4 pages (excluding references) using the Lecture Notes in Computer Science template, and contain the following sections. Up to [20%] marks may be deducted this format guideline is not followed or altered for gaining extra space. The percentages indicate the marks available for each section, which will be awarded together with the relevant code. Each section will be assessed based on, in decreasing importance, its scientific soundness, accuracy, completeness, presentation, formatting and novelty.

- Abstract (10%)
- Introduction (10%)
 - Brief of the motivation and background of the project
- Methods (30%)
 - Description of the implemented algorithms
 - Description of any interesting parameters and designs
- Experiments (30%)
 - Description of a comparison experiment (see The Project section below)
 - Description and justification of the choice of comparison metric(s)
 - The clinical relevance for the chosen comparison metric
- Results (10%)
 - Results and their clinical interpretations
 - Qualitative and qualitative analysis
 - Statistical testing on relevant quantitative conclusions
- Discussion (5%)
- Conclusion (5%)

Data

We will be using a 3D ultrasound image, acquired during a minimally invasive intervention for prostate cancer patients. You can download the original image file from the module repository or via this [link](#). It is in the `gipl` format after unzipping the downloaded file. The gipl image file can be read by supported packages such as simpleitk or medpy, and other visualisation tools such as 3D slicer or itknap. Please include a copy of the data in your submission, for example in npy file, so you do not have to include the package and code that read the image file.

The project: Bilateral filtering on resliced image

This project aims to implement and then use a bilateral filtering method, described in the [paper “Bilateral Filtering for Gray and Color Images”](#), applied on either resampled 2D images or the entire 3D image. The submitted report should contain, but not limited to, addressing the following aspects:

- Implementation of a re-slicing algorithm that obtains an image slice in a non-orthogonal plane, reasonable for viewing prostate gland in 2D images.
 - Why do we need such a re-slicing algorithm in clinical practice?
 - What are the reasonable “positioning” parameters of the re-slicing algorithm?
 - Visualise re-sliced examples in 2D and 3D, with varying positioning parameter values.
- Applying the nonlinear filtering for prostate ultrasound images.
 - Description of the algorithms, for both 2D and 3D versions.
 - What are the parameters of the filter?
 - Analyse the impact due to varying filter parameter values.
 - Discuss the computational performance of the filtering.
- Compare two approaches, “3D-filtering before re-slicing” and “2D-filtering after re-slicing”.
 - Visualise the results from both approaches.
 - Qualitative comparison between the two approaches.
 - What metrics can be used for quantifying the difference between the two approaches?
 - Quantitative comparison between the two approaches.
 - Discuss the potential clinical impact due the observed difference.

The checklist

This is a list of things that help you to check before submission.

- ✓ The coursework will be submitted as a single “cw2” folder, compressed as a single zip file.
- ✓ Under your “cw2” folder, you should have at least a “main.py” Python script, a pdf file “report.pdf”, and the optional “requirement.txt”.
- ✓ The main script runs without needing any additional files, data or customised paths.

Check all the functions/classes have docstring on data type, size and what-it-is for input arguments, outputs and a brief description of its purpose.