# CM20219 – Coursework Part 1 – 2021/2022

Image processing and 2D geometry

Dr Christian Richardt

## 1. Introduction

In this coursework, you will implement functions for filtering images using convolutions, for transforming 2D coordinates, and for warping images according to 2D transforms, as used for example in panorama stitching.

This assignment is worth 10% of the total marks for the unit. The second half of the coursework (on WebGL) is worth 20%, and the exam accounts for the remaining 70% of the total marks.

**Please note:** This is the first year of using an auto-marker for assessing code, so there may be issues and bugs that need fixing along the way. If you have questions about the marking, please ask a tutor. If you think you found a bug, please email me at c.richardt@bath.ac.uk and include the steps to reproduce the bug, the expected result and the actual result. Thank you.

**You can find the lab exercises at https://you.cs.bath.ac.uk/.**

## 2. Assessment

In this coursework, you will be writing functions and code snippets using Python within Jupyter Notebook, an interactive computing environment. It is available on all computers provided by the university (such as those in 1W 3.56), you can install it on your own computer, or use it online, for example via Google Colab.

Some tasks require you to extend a given piece of code, while others ask you to write code from scratch. Please **read the instructions carefully**. If you have any questions, ask a tutor.

This coursework covers the labs in the first half of the unit:

**Week 2:** Introduction to Python and Jupyter Notebook
**Week 3:** Convolution Lab
**Week 4:** Convolution Lab (continued)
**Week 5:** Transforms Lab
**Week 6:** Image Warping Lab (consolidation week)
**Week 7:** Image Warping Lab; deadline for code submission

The first lab, in week 2, will be a brief introduction to the Python programming language and Jupyter Notebook. Should you have any Python/Jupyter-related questions after this session, please ask one of the tutors who are there to help you. The following five labs (weeks 3–7) focus on specific topics covered in lectures: convolutions, 2D transforms and image warping.

You are encouraged to **finish each lab's exercises in the designated week**. In your allocated lab time slot, you can ask questions to the tutors. The last opportunity to get help is in week 7. Note that you can (and may have to) work on this coursework also in your own time.

**The coursework deadline is Friday, 19 November 2021, 20:00**.

**This coursework is entirely assessed by your code submitted to the auto-marker 'You'.**

## 3. Getting started with Python and Jupyter Notebook

Useful resources:

- http://bit.ly/uobcs-python-toc
  Introduction to Python for beginners, entirely in Jupyter Notebooks
- https://www.learnpython.org/
  Interactive Python tutorial, a guide for getting started with Python.
- https://cs231n.github.io/python-numpy-tutorial/
  This tutorial introduces NumPy, a common library for manipulating matrices and arrays, and Matplotlib, a visualisation and plotting library for looking at data or images.

## 4. Learning outcomes

After completing this coursework, and attending the relevant lectures, you should be able to:

- Construct appropriate convolution kernels for common image filtering operations.
- Efficiently compute convolutions, using Fast Fourier Transforms if necessary.
- Devise and apply 2D transformations for animating and aligning 2D objects.
- Warp images using homographies and parametric models, e.g. for lens undistortion.

## 5. Plagiarism

**You must complete this coursework individually.** If you copy code from another student and include it in your project without clear attribution, then you have committed plagiarism.

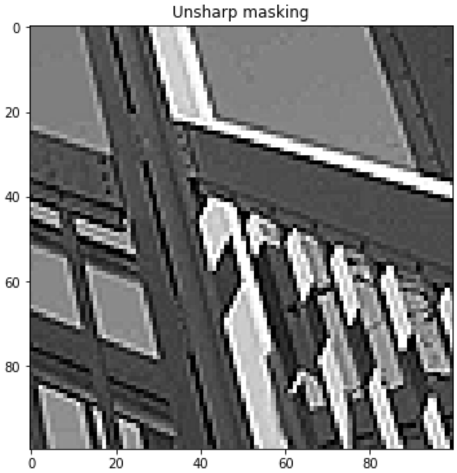
Plagiarism is a serious academic offence. For details on plagiarism and how to avoid it, please visit http://www.bath.ac.uk/library/help/infoguides/plagiarism.html. Undetected plagiarism degrades the quality of your degree, as it interferes with our ability to assess you and prevents you learning through properly attempting the coursework. Consequently, if we detect any plagiarism you will receive zero marks for the coursework and be referred to the Director of Studies for disciplinary action. Such action may affect your ability to continue your studies at the University. Note that properly attributed code, while allowed in your submission, will not contribute towards your marks. The report marks will also only be applicable to sections you have coded yourself.

# 6. Convolution Lab (3.3% of marks for the unit)

This lab implements image convolution of greyscale images. Convolution is a basic image filtering operation that is implemented in:

- many computer vision systems (e.g. for edge detection),
- most image editing programs like Photoshop (e.g. for blurring or sharpening images), and
- many deep learning systems, in particular convolutional neural networks (CNNs).

You can find the Jupyter Notebook for this lab at `https://you.cs.bath.ac.uk/`.



## 6.1. Marking scheme [out of 100 marks]

| Task | Marks | Description |
|---|---|---|
| **1. Basic convolution [40]** | | |
| a | 25 | Implement convolution yourself and produce a result that matches a reference implementation within the central region (ignoring any black border) and possibly shifted by up to 2 pixels in any direction. |
| b | 5 | The result has the same dimensions and datatype as the input. |
| c | 10 | Your function works for general (2*n*+1)×(2*m*+1) kernels; try [–1, 0, 1]. |
| **2. Border handling [20]** | | |
| a | 10 | Extend your convolution function by centring the filtered image and filling in borders by clamping to produce results close to a reference implementation (sum of squared differences < 0.1). |
| b | 10 | Near-perfect match to reference implementation (`numpy.allclose()` is true) for various (2*n*+1)×(2*m*+1) kernels. |
| **3. Image filtering [10]** | | |
| a | 4×2 | Correct convolution kernels for:<br>(1) horizontal gradients,<br>(2) vertical gradients,<br>(3) any diagonal gradients, and<br>(4) unsharp masking. |
| b | 2 | Correct convolution kernel for a normalised 5×5 Gaussian blur (σ = 1). |
| **4. Fast convolution [30]** | | |
| a | 10 | Convolution using FFT that produces results that are close to a reference implementation (sum of squared differences < 0.1). |
| b | 10 | Near-perfect match to reference function for a random 5×5 kernel. |
| c | 10 | Near-perfect match for arbitrary kernels (odd/even width/height). |

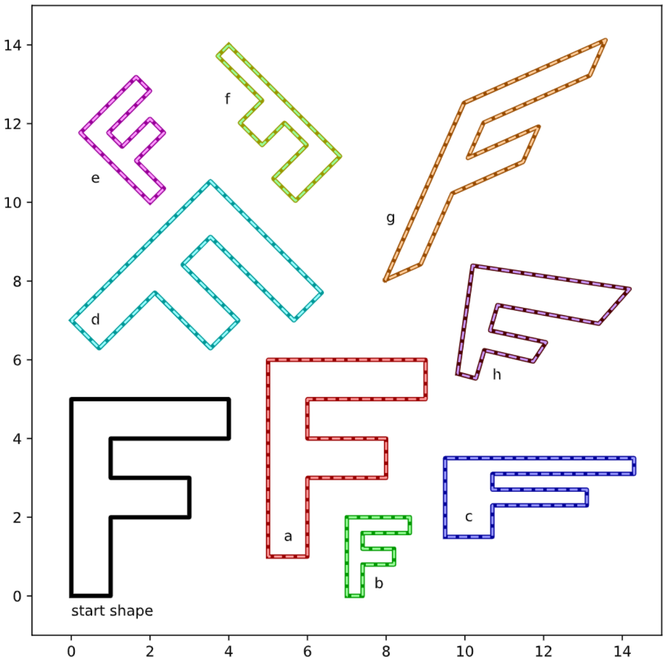
# 7. Transforms Lab (3.3% of marks for the unit)

This lab explores 2D matrix transformations for manipulating and animating shapes.

You can find the Jupyter Notebook for this lab at `https://you.cs.bath.ac.uk/`.

## 7.1. Marking scheme [out of 100 marks]

| Task | Marks | Description |
|---|---|---|
| **1. Basic transforms [20]** | | |
| a | 5 | “`scaling()`” returns a 3×3 homogeneous transform matrix for uniform scaling about the origin. |
| b | 5 | “`translation()`”: same but for translation by a vector. |
| c | 10 | “`rotation()`”: same but for counter-clockwise rotation (in degrees). |
| **2. Compound transforms [20]** | | |
| a | 10 | Combine scaling, translation and rotation into a compound transform. |
| b | 10 | Same but for post-multiplication (row vector × matrix). |
| **3. Articulated motion [20]** | | |
| a | 10 | A second moon is orbiting the Earth as specified in the Notebook. |
| b | 10 | A third moon is orbiting the second moon as specified. |
| **4. Estimating transforms [40]** | | |
| a | 5 | Input shape is transformed to align with thick outline “a” (red). |
| b | 5 | Same  for “b” (green). |
| c | 5 | Same for “c” (blue). |
| d | 5 | Same for “d” (cyan). |
| e | 5 | Same for “e” (purple). |
| f | 5 | Same for “f” (yellowish). |
| g | 5 | Same for “g” (brown). |
| h | 5 | Same for “h” (dark purple). |


a
b
c
d
e
f
g
h
start shape