The University of Melbourne
School of Computing and Information Systems
COMP90086 Computer Vision, 2024 Semester 2

Assignment 1: Image Formation and Filtering

Due:	11:59pm, 24 Aug 2025
Submission:	Source code (in Jupyter Notebook) and written responses (as .pdf)
Marks:	The assignment will be marked out of 6 points, and will contribute 6% of your total mark.

1. Mapping between world and image coordinates [2.5 marks]

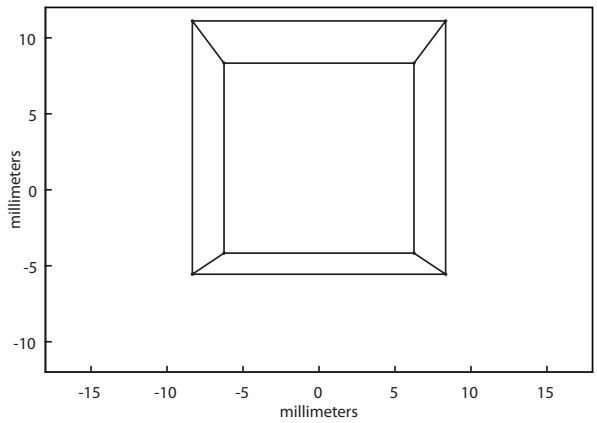

Figure 1: A wireframe cube, imaged by a simulated camera.

The figure above shows a simluated photo of a wireframe cube. This cube was imaged by a simulated pinhole camera with a focal length of 50 mm and a sensor size of 24 mm high x 36 mm wide. The optical centre of the camera is at the point (0,0,0) in the world and projects to the centre of the sensor (the point (0,0) in the figure). The view axis of the camera is the +Z axis in the world. This image has been flipped so that +y in the image corresponds to the +Y direction in the world and +x in the image corresponds to the +X direction in the world. The distance to the closest face of the cube is 90 cm from the camera (the closest face is aligned with the plane $Z_{world} = 90cm$).

In the Jupyter Notebook, we've provided the image coordinates of the 8 corners of the cube and code for rendering the figure above.

1.1. [1.5 mark] Compute the world coordinates $(X_{world}, Y_{world}, Z_{world})$ of the 8 corners of the cube. In your write up, provide the coordinates and show a diagram of this imaging set-up to illustrate how you computed the these coordinates.

1.2. [1 mark] Write code to convert world coordinates to image coordinates for this set-up. Translate the cube in the world (move it up/down, left/right, closer to or farther from the camera) and verify that

the images of the translated cube look correct. Show examples in your write-up.

2. Depth of field simulation [3.5 marks]

Now let's use the image-to-world mappings from part 1 to create a photographic effect in real images, namely simulating low depth of field. An example of an image with a low depth of field is shown below.

Figure 2: Example of an image with a low depth of field. The flowers near the middle of the image are in focus and parts of the scene which are closer or farther away are blurred.

This effect can be created naturally by using a camera with a wide aperture lens. It can also be simulated by selectively blurring parts of the image based on their depth. The amount of blur depends on the length of the vector from the camera's optical centre to the point in the world. Since we model the world with the origin (0,0,0) at the camera's optical centre, this distance is $d = \|(X_{world}, Y_{world}, Z_{world})\|$.

A simple, efficient way to simulate low depth of field for an image I is as follows:

1. Choose a range of distances d which will be in focus.
2. Compute d for all pixels in I and identify which pixels should be in focus.
3. Apply a Gaussian blur or lowpass filter to image I. Call this blurred image I'.
4. In image I', replace the in-focus pixels (identified in step 2) with their original, unblurred values from I.

2.1. [1.5 mark] Implement these steps with the provided images of the University of Melbourne campus. For each image, we have provided an estimated depth map from [1]. The values in the depth map represent Z_{world} in mm. The camera parameters for each image are provided in the Jupyter Notebook template. Show example outputs in your write-up.

Note that your results will not look perfect, even if you implement all steps correctly! For one thing, a real photograph would not have a hard cut-off between depths that are in focus vs. blurred. In

a real photograph, blur would gradually increase for distances away from the focal point, however simulating this exactly in a large image is very slow.

2.2. [2 marks] Aside from the limitation stated above, what other flaws do you notice when you apply this method to the provided images? Critically discuss the method in your write-up, making reference to your image outputs.

Submission

You should make two submissions on the LMS: your code and a short written report explaining your method and results. The response to each question should be no more than 500 words.

Submission will be made via the Canvas LMS. Please submit your code and written report separately under the **Assignment 1** link on Canvas.

- Your **code** submission should include the completed Jupyter Notebook template with your code and any additional files (other than the files we provided) that we would need to run your code. Please include the cell output in your notebook submission if possible.
- Your written **report** should be a .pdf with your answers to each of the questions. The report should address the questions posed in this assignment and include any images, diagrams, or tables required by the question.

Evaluation

Your submission will be marked on the correctness of your code/method, including the quality and efficiency of your code. For efficiency, your image processing steps should use convolution and/or matrix operations (your code should NOT include loops that iterate over all image pixels). You should use built-in Python functions where appropriate and use descriptive variable names. Your written report should clearly address the questions, and include all of the specific outputs required by the question (e.g., images, diagrams, tables, or responses to sub-questions).

Late submission

The submission mechanism will stay open for one week after the submission deadline. Late submissions will be penalised at 10% of the total possible mark per 24-hour period after the original deadline. Submissions will be closed 7 days (168 hours) after the published assignment deadline, and no further submissions will be accepted after this point.

To request an extension on this assignment, please see the FEIT extension policy and follow the steps below:

- **To request an extension of 1-3 days (24-72 hours)**, complete the online declaration form available at the link above.
- **To request an extension of 4-7 days**, please apply for Special Consideration.

Please note that extension requests should be made before the assignment is due. Late extension requests can only be granted through Special Consideration. The longest possible extension for this assignment is 7 days (168 hours).

Updates to the assignment specifications

If any changes or clarifications are made to the project specification, these will be posted on the LMS.

Academic misconduct

You are welcome — indeed encouraged — to collaborate with your peers in terms of the conceptualisation and framing of the problem. For example, we encourage you to discuss what the assignment specification is asking you to do, or what you would need to implement to be able to respond to a question.

However, sharing materials — for example, showing other students your code or colluding in writing responses to questions — or plagiarising existing code or material (including materials generated by chatGPT or other AI) will be considered cheating. Your submission must be your own original, individual work. We will invoke University's Academic Misconduct policy where inappropriate levels of plagiarism or collusion are deemed to have taken place.

References

[1] Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., & Müller, M. (2023). ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth. https://arxiv.org/abs/2302.12288