

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

Assignment 1: Image Filtering and AR Object Rendering

Due: 7pm, 18 Aug 2024
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. Image filtering [2 pt]

In lecture, we saw two types of averaging filters which can be used to blur or smooth an image: the Gaussian filter and the box filter. The box filter is a simple averaging filter that uses a square kernel to average pixels. For example, a 3x3 box filter can be implemented as:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Evaluate the box filter in the frequency domain and explain what impact this filter has on an image. Your write-up should include a visualisation of the Fourier transform of the box filter, and you should reference this figure to support your discussion of the filter's effects.

Note: You may use any **size** of box filter you feel works well to illustrate your points, but we recommend that you use a large **support** (pad the filter with zeros) for visualisation in the frequency domain.

2. AR object rendering [4 pt]

Write a function to paste an object into a photograph at the appropriate scale, given the object's real-world size and its position in the scene. For this problem, use the provided `background.jpg` image (shown in Figure 1) as the background scene.

This image was taken from a camera located 160 cm above the ground with a focal length of 15mm. The imaging surface of this camera is 36 x 24 mm and the optical center corresponds to the center of the image. The camera is oriented so that its optical axis is exactly parallel to the ground plane and the vertical axis in the image exactly corresponds with the vertical axis in the world. You can assume the camera is a pinhole camera with no lens distortion, and the ground plane is perfectly flat.

Your function should accept as inputs:

- a position on the ground plane, specified as coordinates in this image



Figure 1: Background scene for rendering objects.

- an image of an object to be rendered at that location
- the real-world height of that object

The output of your function should be a modified version of `background.jpg` with the specified object rendered at the specified location at the correct size given its real-world height.

For example, suppose you wish to render an object at the center of the bottom edge of the image. **Where is this point on the ground in the world?**

Now suppose you wish to render an object which is 0.4 meters tall in the image, so that the bottom edge of the object is at the bottom edge of the `background.jpg` image. **What should be the height of this object in the image, in pixels?** (For simplicity, assume the object has no depth; it's just a 2D plane parallel to the image plane.)

In your write-up, show a diagram of the scenario described above and explain how you compute the height of the object in this case. Also show the outputs from your function rendering objects at 2 different positions in the scene.

You can assume that the position inputs to your function are always valid (they are always points on the ground plane, within the image bounds). Your function is not expected to handle cases where the object to be rendered in the scene would extend past the boundaries of the image. Your function does not need to work for any arbitrary background image, but it should work for any object. You can assume any objects input to your code are transparent PNGs similar to the provided sample objects.

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 Jupyter Notebook (please use the provided template) with your code and any image files we will 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. 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 (without AAP)**, complete the declaration form at the website above and upload it to Canvas under **Assignment 1 extension request**.
- **To request a longer extension (without AAP)**, please apply for Special Consideration.
- **If you have an AAP**, please request an extension by completing the **Assignment 1 extension request** form on Canvas and uploading your AAP.

Please note that we can only accept extension requests via Canvas up until the assignment deadline. Late extension requests can only be granted through Special Consideration. The longest extension granted on this assignment is 7 days (5 working days + weekend).

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.