

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

Assignment 3: Computing disparity between stereo images

Due: 7pm, 6 Oct 2024
Submission: Source code (in Jupyter Notebook) and written responses (as .pdf)
Marks: The assignment will be marked out of 7 points, and will contribute 7% of your total mark.

Introduction

Finding the disparities between two images is the main step in estimating the three dimensional structure of a scene.

This assignment uses part of the Middlebury dataset, which contains 10 image pairs along with disparity images and camera calibration information.

The assignment starts by looking at the Backpack example in this dataset.



Figure 1: left and right stereo images of the backpack and disparity between them

Your task is to compute an estimate of a disparity map from two images.

1. Baseline code [2 pts]

Run the baseline code implementation in the provided python notebook. Your report should contain:

1. A brief explanation of how the image warping code works
2. Why the result of image warping has lots of black pixels (e.g. to the right of the backpack).
3. A brief explanation of how the disparity estimation code works
4. A description of the errors in the estimated disparity map.

2. Effect of window size [2pt]

For this part of the assignment, you should change the scale parameter used to downsample the image. Using the Backpack example, at scale = 4 and scale = 2, try varying the window size (by changing the halfwin parameter). In your report describe how the size of the window affects the disparity estimate both quantitatively (the numerical performance measure) and qualitatively (describe how the disparity estimate changes).

3. Centre weighted window [2pt]

The window function is the kernel used for the conv2d operation. In the baseline code, this is created using `np.ones()` which considers each pixel in the window equally. Replace this kernel with one that weights the centre pixels more heavily than the edge pixels using a Gaussian weighting function. The sigma parameter should be a quarter of the window size (ie half of the halfwin parameter). Repeat the experiments for Q2 and determine if a weighted window needs a different window size for optimum performance.

4. Obtain performance across the whole dataset [1pt]

Using the best performing window size at scale 2, run your code on the remaining 9 examples in the dataset and describe the performance quantitatively, giving your results in a table.

Conclude your report with a discussion of the main errors made by this system and how these might be addressed using a more complex approach.

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 3** link on Canvas.

- Your **code** submission should be just your Jupyter Notebook (please use the provided template) with your code. Please include the cell output in your notebook submission if possible. DO NOT INCLUDE THE DATASET.
- 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 3 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 3 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.