Homework 02: Filtering

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2025-09-08

Due: 2025-09-15, 11:59pm, on Canvas

General Instructions:

- You should solve the homework and submit your report **individually**. Identical submissions will receive a grade of zero.
- Getting help from others or checking your answers with other students (not the TAs) is okay and encouraged.
- Ask any questions on **Ed Discussion** (instead of emailing).
- **Before** the homework due date, TAs are strictly prohibited from **pre-grading** your homework. Do not expect the TAs to help you verify if your answers are correct or give you the problem solution.
- **After** the homework due date, if you do not know how to solve a problem, reach out to the TAs. They will walk you through the solution and help you understand it. Note that homework solutions will **not** be posted because some problems will be used in next year's class.
- **Exams** may contain questions related to homework, so make sure you learn how to solve the homework problems correctly.
- The deliverables are outlined for each problem, and you should carefully **follow the instructions**. Failing to follow instructions will result in **points being subtracted**.
- You will submit a **single PDF** file to Canvas as your homework report. The PDF must contain your **answers** and any requested **outputs** (e.g., printouts, snapshots of code, or GUIs). If requested, follow the instructions specified by the problem to provide your **code** (e.g., in a compressed .zip or .tar file) in addition to the PDF file.
- **Grading:** Each homework in this class will contribute **5pts** to your final grade (there will be 12 homework assignments, each 5pts, leading to 60pts for all assignments). A detailed grading **rubric** will be posted on **Canvas** after the homework due date. Any bonus points will be added to your overall course bonus points, which will be added to your final grade.
- **Late submission:** Late or missed submission will not be accepted and will receive a grade a zero. Any excused absence must be documented and disclosed to the instructor (extensions will be granted on a case-by-case basis). Three or more missed homework lead to an INC grade.

Homework 02: Filtering

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2025-09-08

Due: 2025-09-15, 11:59pm, on Canvas

EXERCISE 1 (5pts) – The objective of this homework is to learn more about image filtering. Do **NOT** use generative AI to solve this homework; otherwise, you will not learn to code!

Steps:

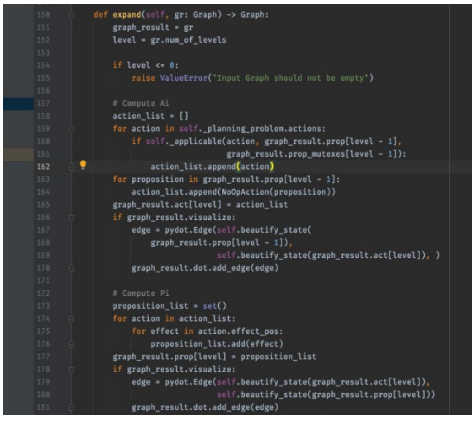
- Load an image of ***yourself*** (e.g., the one used in HW01). Convert it to grayscale and appropriate type (e.g., float32) if needed.
- Apply the following 3x3 filters ***without*** using an existing implementation by implementing the filtering equation yourself. You need to implement the filter yourself using `for` loops. If you use existing filtering implementations in Python libraries (such as `correlate` and `convolve` filters), you will not get any points.

$$f_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \qquad f_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad f_3 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

  Hint: consider downsampling your image for speedup. Having 50 to 100 pixels in each dimension should be sufficient.
- Display the filtered image. You may need to adjust the pixel values/ranges (e.g., if they are negative) and select a colormap so that the output image looks correct.
- Explain what each filter does. If the filter has a name, mention it.

Deliverables:

- Snapshot of your entire code, e.g.:
  Snapshot of your input (grayscale) image
- Snapshots of your filtered images for filters f_1, f_2, and f_3
- Explanation of what each filter does and its name (if discussed in class)



```
def expand(self, gr: Graph) -> Graph:
    graph_result = gr
    level = gr.num_of_levels

    if level <= 0:
        raise ValueError("Input Graph should not be empty")

    # Compute Ai
    action_list = []
    for action in self._planning_problem.actions:
        if self._applicable(action, graph_result.prop[level - 1],
                            graph_result.prop_mutexes[level - 1]):
            action_list.append(action)
    for proposition in graph_result.prop[level - 1]:
        action_list.append(NoOpAction(proposition))
    graph_result.act[level] = action_list
    if graph_result.visualize:
        edge = pydot.Edge(self.beautify_state(
            graph_result.prop[level - 1]),
                          self.beautify_state(graph_result.act[level]), )
        graph_result.dot.add_edge(edge)

    # Compute Pi
    proposition_list = set()
    for action in action_list:
        for effect in action.effect_pos:
            proposition_list.add(effect)
    graph_result.prop[level] = proposition_list
    if graph_result.visualize:
        edge = pydot.Edge(self.beautify_state(graph_result.act[level]),
                          self.beautify_state(graph_result.prop[level]))
        graph_result.dot.add_edge(edge)
```