Guide to CIS Programming Assignment

THE REPORT

October 2025

1. Overview of the PA Report Grading Rubric

The report for the programming assignment is worth a significant amount of points (45 points) in the assignment, so we are looking for well structured, quality reports with proper documentation of your program and results summaries. The report is expected to be divided into sections, with the following points assigned to them:

- Overview: Overview of the problem and context of the assignment (1 pt)
- Mathematical approach: mathematical equations and derivations behind the approaches used (7 pts)
- Algorithmic approach: algorithmic methods and variables translating mathematical approaches to programming execution (7 pts)
- Overview of structure: overview of the functions and structure/hierarchy of the overall program (10 pts)
- Validation Approach: proof of method testing with steps (10 pts)
- Results: evaluation and results of debug datasets, unknown results, and work division (10 pts)

The closer you follow this structure, the less likely a grader might miss discussion relevant to a different section while grading. You may find using a platform like Overleaf useful for citation management as well as for equation writing.

PLEASE STRUCTURE YOUR PROGRAM REPORT BY SECTION INSTEAD OF BY PROBLEM. IT IS HELPFUL TO STICK TO THE STRUCTURE GIVEN BELOW.

The following sections can serve as a report template and will more thoroughly specify what would be a good result for each section.

For Programming Assignment 2, please still include mathematical equations and algorithms for methods already discussed in PA1 (ie. registration, pivot calibration).

2. Summary of the Problems to Be Solved

We are looking of a description of the problem scenario, including the methods (eg. calibration, registration) and context (eg. navigation system). It may be helpful to put the problem scenario diagram, but it is not necessary for full points. If you defined any of the variables here, they do not need to be redefined later on in the report.

What we are not looking for is just a list of the specific goals given in the homework in the exact same format given. This can be a brief section as long as you address the what and why.

3. Mathematical Approach

The mathematical approach needs to contain **all of the derivations and equations** behind the mathematical formulas used in your programming assignment. We are expecting citations for these equations, including Dr. Taylor's lecture slides. We expect this for all equations (yes, pivot calibration too). We would prefer proper/complete citation notation as well.

Equations not typeset into the text are best. As is standard practice, please make sure the variables in your equations are defined. It would be nice if different approaches are in different subsections, but this will not be penalized.

Make sure to remember to include methods for corner cases. Specifically for 3D point set registration, please mention what to do when det(R)=0 (ie. zero singular value case).

For PA2 we would generally like to see the following topics/approaches covered in this section:

- 3D point set to 3D point set registration
- Pivot Calibration
- Distortion correction
- Frame transformations/equations to get desired variables (eg. frame transforms to get C_exp, F_reg, b_j, v_j)

4. Algorithmic Approach

The algorithmic approach section ties your mathematical approach to the program structure and execution. Here we are looking for pseudocode and variables definitions for implementations of the mathematical approaches and other key functions, as well as how they all work/flow together. Inputs and outputs can be included here or in the program structure. You need to include what programming language was used here as well as any non-standard libraries used (NumPy is not a standard library). Standard functions like math packages for points, and transforms do not need to be described in this section.

From your algorithmic approach, there should be enough detail that we can generally reproduce your functions ourselves. If any assignment-specific variables haven't been defined yet, they should be defined and used here. You can break up your algorithmic approach by the functions implemented in your program, just make sure that it isn't a pure copy of your code.

A general rule to use if you are trying to decide what to include in this section is that if it was explicitly mentioned in the suggested procedure, it is likely an important enough function/process to include here. If it was in the mathematical approach section, it definitely needs to be here.

Algorithm 1 Overleaf algorithmic

Input parameters
Output description
if you use this method **then**
    Overleaf algorithmic can make very nice looking algorithm structures :)
else
    Just make sure you are describing the steps.
    Numbered lists would also work too.
end if
return

5. Overview of Program Structure

This section should show the complete overview of your program structure so that someone could open up your submission and understand what all the functions are, where everything is, and how the functions/-classes/scripts all call each other.

Charts (hierarchy charts, flow charts) are very useful visualization tools for your program structure and will guarantee 5pts on their own. If you don't have a chart showing hierarchy, the function hierarchy needs to be discussed/shown in another form.

We want to see everything, so list all of your functions, including your helper and data processing functions. If you did not mention inputs and outputs earlier, you need to mention them here. You can include "code blocks" if you'd like (shown below) where the inputs and outputs are both in the same line as the function. However, you then need some description of what each function does before/after.

```
%% Data Reading Functions
function output = myFunction(input)
function output = myFunction(input)
```

Please also include a brief description of the output of the program.

6. Validation Approach

In this section, we want to see how you are testing/checking your methods to make sure that they work. We are specifically looking for unit testing and result analysis, which means for relevant functions/methods, we want to see:

- What unit tests/functions you made and for what part(s) of the program
- your testing process (steps? did you generate fake data? how much did you test on? etc)
- results (ie. RMSE error, numerical proof that your function does work).

We are looking for a level of detail in these testing process descriptions so that someone reading your report could reproduce your methods. In addition, whenever you are evaluating error, please include the equations that you used so we know your error calculation method.

While certainly not required, plots can be nice to showcase some of your validation results (eg. mis-aligned $\rightarrow$ aligned point clouds).

7. Results

The results section has two main parts: the debug dataset, and the unknown data set.

For the debug dataset, we are looking for evidence of analysis between your debug outputs and the provided debug outputs. This can be in the form of tables showing the error(s) calculated between your output and the provided outputs, such as error in each component of the post position and average C_expected error for PA1. Please include the equations and methods you used to calculate the error. We would like to see analysis using all of the debug data sets provided to you for the assignment.

If there is significant deviation between your output and the given output, please give a reasonable and thorough explanation as to why/where it could be coming from. At the end of the PA document is a table showing what debugs have noise, distortion, or jiggle.

Finally, please output the results of the unknown data set in a tabular form.