Reflection and Traceability Report on ESF

Alaap Grandhi

1 Changes in Response to Feedback

1.1 SRS and Hazard Analysis

Feedback	Source of Feedback	Resulting Change
Variables not consistently wrapped in dollar signs (did not show up italicized)	Professor Smith	Wrapped the variables in dollar signs in all locations where they were missing
Symbol Table entries missing for variables describing the shape of other variables (e.g. the number of classes)	Professor Smith	Added the missing entries to the symbol table
Incorrect spacing for the true positive, false positive, and false negative symbol table entries	Professor Smith	Wrapped these entries in \mathit{} as suggested
Saying that none of the sensors should completely cut out at any point in time seems to contradict introduction	Professor Smith	Clarified that this just means that the sensor readings should always be input in the same format, even if they are distorted or blank.
Goal statement two is missing the goal statement one's trained model as input	Professor Smith	Added this into the goal statement
Assumptions need traceability to where they are used	Professor Smith	Added a short statement at the end of each assumption denoting where they are used with links
The variable $\mathbf{P}_\mathbf{c}^*$ was used but not defined in the Symbol Table (needs to be defined)	Professor Smith	Since that variable is only defined and used in that Theoretical model (TM:Intrin), I simply added a '$\triangleq$ ' to make this clearer
X_c, Y_c, Z_c were used but not defined in the symbol table	Professor Smith	Added these symbols to the symbol table

Placement of certain variables (e.g. P_c in TM:Intrin) led to confusion on what they were referring to	Professor Smith	Moved the variables to more appropriate locations where necessary to enhance clarity
Specifying gradient descent is less abstract and requires justification (needs to be presented as a constraint)	Professor Smith	Since ADAM is a gradient descent method, I specifically mentioned gradient descent in the ADAM constraint (C3) to clarify why it is chosen
The controllable parameters in TM:AP are presented as a set, but this choice is unclear (why not a vector?)	Professor Smith	This was changed to a vector in the documentation, with additional specification that all parameters are in R
The mAP metric is formatted differently in different places	Professor Smith	I formatted all of the mAP references as plain text rather than with the math italicization
The rationales are not linked to by the items they are justifying (meaning the reader does not know that certain items are justified until they reach the rationale section)	Professor Smith	Added links to the rationales where relevant

1.2 Design and Design Documentation

1.2.1 Module Guide

Feedback	Source of Feedback	Resulting Change
The coloured arrows in the use hierarchy might cause the reader to look for some hidden meaning	Professor Smith	Added a statement (as suggested) specifying that the arrows are just coloured to make the traceability easier to see
There are placeholder module entries with empty fields that do not specify anything	Domain Expert	I removed the placeholder entries
The modules described in the Module Guide are not linked to their specifications in the Module Interface Specification document, causing a disconnect	Domain Expert	Added cross references between the MG modules and their specifications where relevant (the actual links only work in some pdf editors/viewers)

1.2.2 Module Interface Specification

Feedback	Source of Feedback	Resulting Change
The entire SRS URL is shown when it should just be linked	Professor Smith	Replaced the full link with a hyperlink
Constants are defined but no values are provided for them	Professor Smith	I added default values to constants but specified that they may be changed in future revisions
Abstract Objects are called modules in the state variables and exported access programs sections	Professor Smith	I fixed this so that ADTs are specifically called Abstract Objects when referring to singular instances
The usage and purpose of the state variables in the training are unclear	Professor Smith	Clarified that these state variables are abstract objects and clarified that they are modified and used in-place in the training module
The local functions in the training and inference modules describe implementation, not specification	Professor Smith	I removed these local functions and replaced them with exported access programs that instead focused on state transitions and output rather than implementation
Certain variables are not defined or typed	Professor Smith	Added in the Symbol Table from the SRS to clarify variables
Inference module does not have a transition, nor an output (does not do anything)	Professor Smith	Added state transitions and output to the inference module to make its purpose clearer
Access routine schematics are missing inputs (hinders clarity)	Professor Smith	Specified inputs in each access routine schematic
The logger modules is not referenced anywhere	Professor Smith	Added cross references to the logger and any other referenced modules throughout the document

ADTs are called module when they should be called template	Professor Smith	Fixed this throughout the document
Constructors are called init when they should be called the module/ADT's name	Professor Smith	Fixed this throughout the document
Computer Screen environmental variable is missing a type	Professor Smith	Specified that the type is 2D sequence of RGB tuples as suggested
Format of the yaml file is unclear to the reader	Professor Smith	Added a hyperlink referencing a sample yaml file as suggested
The integration of the OpenPCDet and Equivariant Layers Modules in the Model Module is unclear	Domain Expert	Added an in-depth description in the model module's constructor to clarify this (linking the BEVFusion paper for further reference)

1.3 VnV Plan and Report

Feedback	Source of Feedback	Resulting Change
Having two primary objectives and one secondary objective is confusing	Professor Smith	I changed the wording so that there is one primary, one secondary, and one tertiary objective as suggested
Booktabs tables would look nicer	Professor Smith	I have not actually implemented this one yet as I am still learning how to format booktabs tables well (these tables are booktabs tables and part of my learning for this)
SRS checklist reference is missing a citation	Professor Smith	Added a hyperlink to the SRS checklist
SRS verification plan would benefit from being better structured	Professor Smith	Completely overhauled this section with inspiration from Cynthia Liu's MEng report
VnV verification plan contains detail that would be better suited for the Unit testing section	Professor Smith	Removed this part and streamlined the VnV verification plan
Implementation verification plan is missing a description of the code walkthrough structure	Professor Smith	Added a more detailed description for how code walkthroughs will be conducted
Software validation plan is describing verification, not validation	Professor Smith	Fixed this talk more about how to determine if the requirements translated to a viable product
Test cases are lacking specificity	Professor Smith	Added a more detail where necessary, indicating where data is coming from, what file formats are being used, etc
Test for installability is missing	Professor Smith	Added a new testing procedure for installability

Responsibilities for each member of the VnV team are ambiguous	Domain Expert	Added a new deliverables column to the VnV team table to clarify the responsibilities for each member
Nonfunctional testing could be extended to include performance testing	Domain Expert	Since performance testing is very setup dependent, I added a small blurb on how this could roughly be done but not a specific test
Tests cover expected scenarios but not unexpected input	Domain Expert	Added more tests for edge cases as suggested
Unit testing section is largely missing	Domain Expert	Added in the unit testing section

2 Challenge Level and Extras

2.1 Challenge Level

Research project

2.2 Extras

None

3 Design Iteration (LO11 (PrototypeIterate))

To be completely honest, I have not made many design iterations at all. This is mainly because I have not really had the time to finish my implementation of the code itself and so I cannot iterated on the code until I finish the first draft. I suspect that I will make many changes to the type of equivariant layers I use and the transformation groups I explicitly consider for those layers, but I cannot do this until I get full results to begin with.

4 Design Decisions (LO12)

While time limitations did affect my design decisions in terms of preventing me from conducting a full set of experiments (comparing to several baselines), I think that I should have considered the impact they have more at the start of the project. I think I bit off a bit more than I could chew here in that the extent of the coding and documentation needed for this project were beyond what I could finish in the course's duration. In particular, I did not consider that other factors such as TA'ing would take up so much of my time. If I had budgeted my time better with this in mind, I could potentially have gotten some quantitative mAP results in time for the final documentation.

I do not think that my assumptions or constraints really affected my design decisions as I chose them to be in line with state-of-the-art methods.

5 Economic Considerations (LO23)

There may indeed be a market for my product/software for autonomous vehicle perception systems, but I do not intend to market/sell it for that purpose. I would rather release it open-source for researchers primarily, as I hope that it could lead to further advances in state-of-the-art techniques. I aim to attract users for this software by writing up a conference paper for it once I finish experiments and presenting in front of other researchers in the field.

6 Reflection on Project Management (LO24)

6.1 How Does Your Project Management Compare to Your Development Plan

I do not think that I followed my Development plan well to be honest. I was so focused on just finishing the documents one-by-one and overloaded with marking work that I neglected the development process a bit. I built a habit of doing things closer to the deadlines rather than gradually through the semester.

6.2 What Went Well?

I think that the actual course, design process, and design discussions really helped me learn more about designing scientific software. Once I got the hang of writing the documentation, it was kind of cool to see how the structure forced me to clarify ambiguities about the problem in my own mind. It was also super cool to see how the set of documents worked as a sort-of level-by-level pseudocode for the end implementation, providing a clear point of reference for when I wanted to revisit aspects of the problem. However, I think that the effectiveness of this was somewhat limited by the short duration of the semester and that it would work better with a longer course or if I were working as part of a group. I think the focus on traceability also really helped with links between documents and sections as it made quickly jumping between concepts much easier. The use of latex for all of the documents also made structuring them easier.

6.3 What Went Wrong?

In addition to the tight time constraints that I have mentioned above, I think the structure of the course was a bit mismatched with my project. Since I chose a research heavy project, I think that the waterfall-style of the course delayed the actual coding too much. Had I followed a more agile-inspired approach (coding in increments throughout the semester alongside documentation), I think it would have been easier to get results and conduct tests by the end. With the heavy documentation focus, I found it a bit difficult to find time to mess around with the code for extended sessions. With a project like mine where this kind of extended coding is necessary to make progress, I think my final results may have suffered a bit (although this is also due to not making full use of the time).

6.4 What Would you Do Differently Next Time?

I think that if I were to do the process over again, I would follow feedback and close github issues as they are given rather than in a bunch at the end of the course. This way I could also follow my VnV plan sections alongside the design process (verifying each document as it is completed). This would in turn free up more time at the end of the course for pure implementation which I was stressed for time with now.

7 Reflection on Capstone

7.1 Which Courses Were Relevant

There were several undergraduate machine learning and computer vision for robotics courses that were helpful here. This is because they gave me a solid foundation on which to figure out the mathematical models underlying this problem. They also made it much easier to dissect and understand the baseline architectures.

7.2 Knowledge/Skills Outside of Courses

Outside of courses, I did a PEY in my undergrad that focused on Autonomous Vehicle Perception. I also published two papers during the course of that internship. This gave me a strong foundation with which to understand the task and the setting for this project.

The main skill that I had to acquire for this course was learning how to write documentation for scientific software. Having done my undergrad in Engineering Science with a focus on machine learning, I learned about different models and mathematical concepts but never really had to write any documentation. As such, this course was really new to me for the most part. I think it was made a lot easier by the lectures though and I feel a lot more confident in documentation now than I did at the start of the semester.