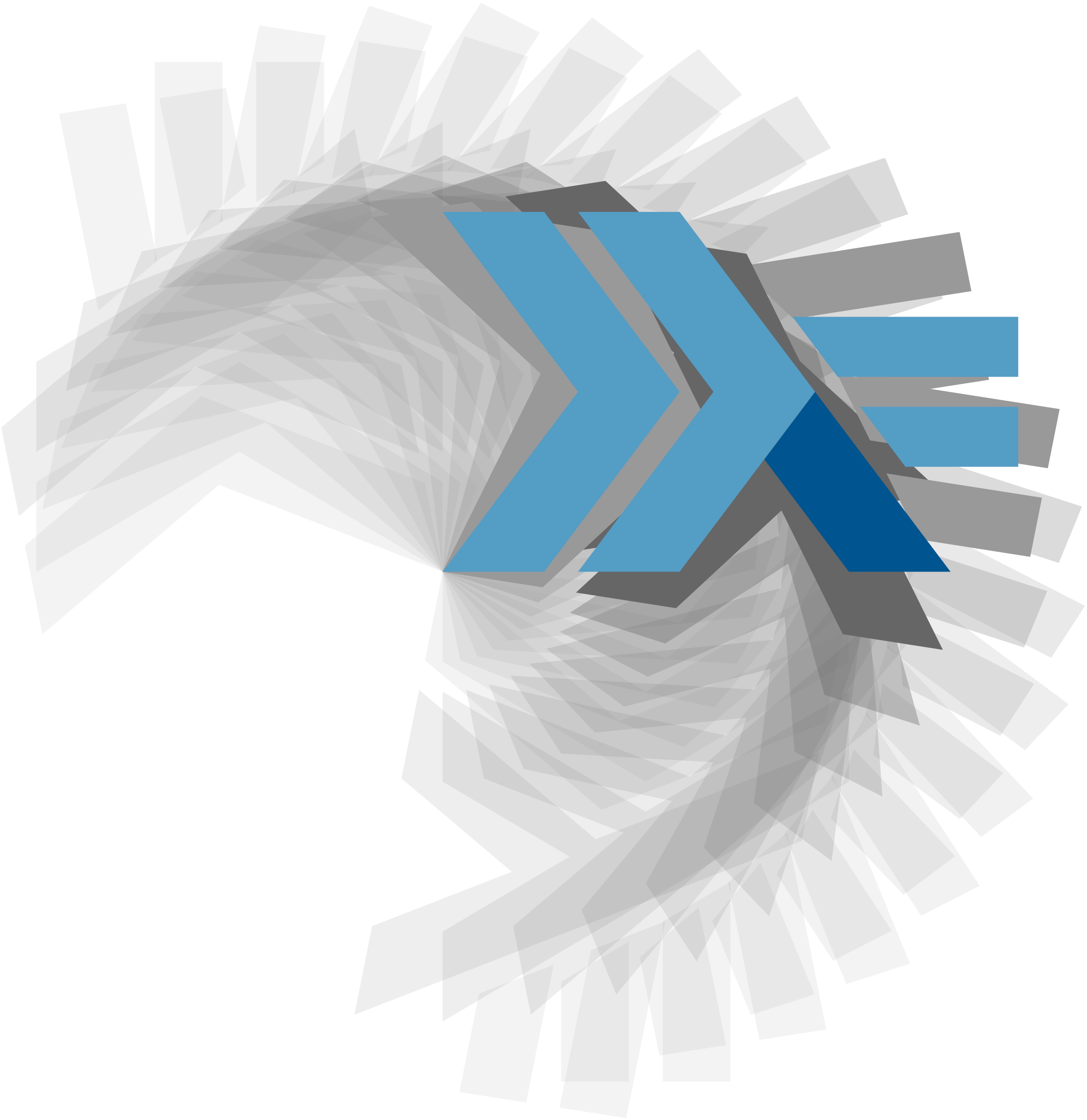

Third Year Group Project

Team Agreement

Jordan Hall, Bartlomiej Cieslar, Robbie Buxton,
Ethan Range, Charlie Lidbury and Oliver Killane

Project Goals

What is the overall goal of your project?

The goal of this project is to develop a toolkit for the auto-labelling of road-markings for autonomous vehicles. Our client Ghost Autonomy will use this is to create a training set of ground truth data for their machine learning models.

We will build this toolkit upon fused-effects, an algebraic effects library for Haskell. It is based on our supervisor's work, providing an elegant, composable system. To implement probabilistic models, we will use ProbFX, a probabilistic programming library built with fused-effects. To achieve our goal, we will need to extend and modify the existing ProbFX library, as well as create a library for computer vision operations which interacts seemlessly with ProbFX.

As an adjacent goal, the existing documentation for fused-effects and ProbFX is somewhat lacking, and so any improvement in these areas would be beneficial.

Team Composition

Team composition: why did you choose the team members you did? Do you have a good balance of skills? Have you allocated any specific roles to particular members?

When selecting members, we looked for a number of qualities in all members; strong programming ability, passion for specific specialities which we could then integrate into our topic, and commitment to putting in the hours required.

More specifically, each team member had notable qualities which influenced their selection:

Bartek

- Functional programming skills
- Mathematical background
- Organisation skills
- Experience working in industry (Nvidia, Meta)

Charlie

- Functional programming skills
- Knowledge of type theory & algebraic effects
- Software engineering & architecture
- Good knowledge of programming language design (beneficial for ProbFX as it is a DSL)

Jordan

- Experience in machine learning & computer vision
- Functional programming skills
- Experience in software architecture design

Ethan

- Functional programming skills
- Administration & Organisation skills (particularly with previous group courseworks)
- Proficiency in probability & statistics

Oliver

- Functional programming skills
- Proficiency in probability & statistics
- Software engineering fundamentals

Robbie

- Research experience from his UROP
- Group Management & Communication
- Experience with computer vision

All members of the group have worked with each other before in some capacity, and so are we were able to vouch for the abilities of some members based on past experience.

We set out to have a good balance of technical skills in the areas that we want to develop the project in and are passionate about: Machine Learning, Computer Vision, Language Design, Theoretical Research and Functional Programming. This breadth gave us the flexibility to explore many potential project areas. We did however also consider the balance of soft skills within the group as well, which will be crucial to ensure that individual member's technical abilities can work well together. We sought members who could provide organisation, design and people-management skills to enhance the efficiency with which the team will work.

We have deliberately shied away from assigning roles to members of the group, as we expect that the dynamic of the project and team member's strengths will naturally give way to roles, which could conflict with pre-assigned roles. We are keen not to elevate one person to the role of the leader.

As the scope of the project is still quite broad, we feel that strict assignments to specific areas of the project may limit the amount of experimentation possible, and so are unhelpful for now. That said, various group members have expressed particular interests in specific project areas, which we have discussed in the Project Plan.

Time Commitment

> *How many hours per week have you each committed to work on the project?*

As a group, we have made a conscious decision to not commit to exact hours for the project due to the diversity of timetables and working styles within the group. Furthermore the differing densities of workloads for the project (e.g complex model creations vs build and boilerplate) reduces the meaningfulness of time committed as a metric. Therefore we will use a more holistic approach - evaluating task priority & contribution size in context as a group.

That said, we do intend to roughly abide by the recommendations of the module leaders, of 250 hours per person over approximately 10 weeks of development. There therefore exists a baseline expectation of around 25 hours per person, per week. However both the members of the group and our supervisor recognise that we are attempting a challenging project with a large possible scope, and so we will likely require a greater time commitment than suggested. We have therefore set a guideline target of 30-35 hours per week spent on the project, for a total of around 325 hours over the course of the project.

Team Management

> *Are you adopting a specific software engineering method or process in your team (e.g. Scrum, XP, Kanban)? What does this mean in practical terms for your team and the way that you will work?*

Our project concerns a conceptually difficult topic. As such, it will likely follow a cyclic process of research, development and testing, which would make splitting work into tasks too far into the future challenging and potentially unhelpful, as the requirements may suddenly changed based on new research. The Scrum methodology is therefore not particularly suitable for us as it requires full knowledge of the work for a specific sprint beforehand, and so is more suitable for projects with shorter feature implementation loops and more rigid scopes. We also discounted another methodology we considered, Extreme Programming, for a variety of factors; the scope of this project means that it is infeasible to have everybody knowledgeable in every area, the quality assurance focus of XP does not fit with our proof of concept requirement by the client and so may hamper experimentation, and as mentioned before, a time-boxed system will likely not suit us.

Therefore, we have chosen to follow the Kanban methodology, which is significantly more flexible and continuous in its nature than the aforementioned methods. Its dynamic use of a task board makes it ideal for us, as it allows us to tweak our workflow as necessary. It also lets us group tasks into goals, which we can create on a per-milestone basis, but also update as requirements change.

In practical terms, this will mostly involve all work undertaken being translated into tickets and placed on a shared Kanban board, to provide the visibility required. These tickets can then be assigned users and moved to indicate the progress of the task, which will allow us to keep work moving at an acceptable pace. The continuous nature will allow us to add or modify tickets as we see fit, which is likely to occur, as a research task being completed may well spawn additional implementation tasks which can be added to the task list. While much of this can be done asynchronously, we will use our meetings to ensure that the board is up to date and that any disagreements with tasks are resolved.

Technology

> *What (if any) technology will you use for a) source control, b) tracking the progress of work in your project, c) communication amongst team members, d) storing and sharing documents*

- **Source Control - Github**

  For source control, we have set up an organisation on GitHub which will contain the repositories we will be working on. Through the tools provided by Github, such as Github Actions, we will be able to ensure and enforce standards for the organisation, hygiene and protection of our code base. These standards can all be automatically applied through CI pipelines, removing overhead from development.

- **Progress Tracking - Jira**

  In order to track the overall progress of the project, we looked for a tool which would allow us to define, allocate and track tasks in a fluid manner, giving us specialised tools for development work while also providing enough flexibility to include research tasks in our tracking. We decided to use Jira, as it provided a reasonable UI experience which some of the group members had experience with, as well as good intergration with Github, but still allowed sufficient flexibility.

- **Communication - Discord**

  As communication is a top priority in such a large and complicated project, we have deployed a Discord server for the project. This will allow us to create separate channels for discussing different aspects of the project in an organised manner, and provides a place to host video meeting in the event that not all team members can be physically present. Having a single, central communication platform will help us avoid fragmentation of information, and provide a permanent record of discussions.

- **Documents - Notion / Google Workspace**

  For storing and sharing documents, we will utilise Notion, a general note-taking application, as our central repository of information. With Notion, we can organise files, and discussion surrounding files, in the same place. We do however plan to use other tools where more appropriate, such as the Google Suite of Office tools due to their superior concurrent editing support, though external documents will all be linked to from the Notion to ensure that information remains centralised.

Meeting Schedule

Have you set up a schedule for regular team meetings? When will they be?

As of now, there will be regular meetings with our supervisor held each Monday, and a team meeting held on Friday without the supervisor. We hope that these regular meetings will help track progress in the project and control the scope of the project. Of course, the dates/times of these meetings are subject to change with the availability of group members and our supervisor.

These meetings will of course be supplemented through continuous communication on the aforementioned Discord server.

Team Rules

Have you set any team rules?

While we have not created extensive rules prior to starting the project, as we would instead prefer to adapt as we discover what works best for our group, we have laid some initial ground rules:

1. Everyone has to attend the weekly Monday and Friday meetings.
2. Infrequent absences from the 2 weekly meetings is acceptable provided that this is communicated in advance.
3. We will enforce a strict coding style for Haskell (using Hlint), commenting and documentation.
4. We will have a mandatory paper reading list because of the research component of this project
5. There will be strict Git rules about naming commits, merging, passing the pipeline and code review, to be set out in a future document.

Contribution Assessment Scheme

> *All teams will complete a simple peer-review exercise at the end of each iteration. You'll need to say whether each team member's contribution, matches, exceeds, or falls short of the team's expectations. What will be the criteria by which you judge each other's contribution to the project?*

We expect that communication will be a key metric. There is a lot of diversity within the group with regards to obligations, modules and future plans (e.g. 2 BEng 4 MEng, some applying for further study, some applying for industry roles, individual projects to be undertaken, et cetera). Therefore the likelihood of everybody being able to attend all meetings is low. As such, we will need to effectively communicate and coordinate for meetings, especially with a timezone constraint from our Ghost Autonomy contact being based in Australia. We will assess contribution based not just on attendance, although this will need consideration, but also on commitment and reliability to showing up when agreed and working when agreed.

Due to the uncertain nature of our roadmap, difficult tasks completed will be judged relative to the complications experienced during the task, and how well people judge their own abilities. For instance, one might fall short of their goals because they misjudged their own ability, or one might fall short because of huge, unforeseeable complications. We need to differentiate between these scenarios to accurately identify problems within the group and recognise efforts made.

Project Plan

> *Give a brief overview of your initial project plan.*

Writing complex Haskell programs often requires the use of monads, which do not compose very well together. An alternative way to construct Haskell programs, allowing for composition, is by using algebraic effects and effect handlers. While Haskell does not natively support aleabraic effects, they can be implemented as monads combined with monad transformers. The fused-effects library is a framework providing algebraic effects in Haskell.

Probabilistic programming is a paradigm of manipulating and combining probability distributions, and allows probabilistic systems to be expressed and manipulated. ProbFX is a library created by a former masters student of Nick Wu to do probabilistic programming in Haskell, built on top of fused-effects.

Our supervisor, Nicolas Wu, was approached by a contact at an autonomous vehicle start up, Ghost Autonomy, looking to use ProbFX for the creation of probabilistic models to detect road markings from input images. This would allow them to label training data, slower than real time, which could in turn be used to train faster machine learning models. Nick in turn contacted us about this project.

In order to prototype this for Ghost Autonomy, we will need to create a framework on top of fused-effects, building on Nick's research, to allow for the image processing and labelling pipeline to be run. The existing ProbFX library will allow us to define and update probabilistic models through Bayesian inference, however we will need a computer vision library built on fused-effects to interact with ProbFX. This effectful computer vision library will be one of our deliverables.

The first milestone will involve creating an extremely simple version of this entire pipeline. We will add the ability to provide images of a line on a blank background to the pipeline, use basic computer vision to extract the lines from these images and make a model which detects the equations of lines on the blank background. This will lay the foundations for us to model the probability of these lines being road-markings in ProbFX, updating the model with techniques such as Markov Chain Monte Carlo sampling and Bayesian Inference. From this foundation we hope to be able to create a full pipeline for model creation and refinement.

Within the group, our strengths align well to the different parts of the projects, and so we will initially assign tasks to align with them. Robbie and Jordan will focus on the computer vision part of the pipeline. Ethan and Oliver will be concerned with the statistics underpinning the model. Charlie and Ethan will be involved with effect handlers and building on top of fused-effects. Bartek, Oliver, and Charlie will be implementing the effectful computer vision library.

⚠ Project Risks

What are the main risks to the success of this project?

Issue	Risk	Description
Dataset Access and Quality	Low	This project is fairly contingent on having access to a large and high quality dataset of road views taken from inside cars. We expect to get access to one from Ghost Autonomy however if they do not provide this data then we may need to search for one on the internet/compile it ourselves.
Model Feasability	High	We need to be careful when building our computer vision model not to overscope and try and build something infeasible in the timeframe. It will be important to keep in mind that the point of the project is not to build the best vision model possible but rather demonstrate the capabilities of ProbFX and whatever other infrastructure we create.
Divergence	Low	It will be important to keep in contact with Ben Lippeimer so we don't diverge too much from what Ghost Autonomy wants to see examples of implemented. We need to make full use of his extensive knowledge about computer vision.
Alignment of Goals	Medium	It will be important to balance the requirements of the company and the university as these are not always aligned. This will be somewhat mitigated as the project is split into two (fused-effects Framework vs The model we develop using it).
Time Constraints	Medium	During term there are several large courseworks that could potentially divert time & focus from the project. For example the *70050 Introduction to ML* labs, or *60007 Concurrency* programming courseworks. In order to mitigate this risk we have and will continue to push for an ambitious Milestone 1, in order to provide a deadline safety buffer.
Model Quality	High	In order to be a useful demonstration of our work, the final lane-line detection model needs to be sufficiently accurate to be useful for Ghost Autonomy. As we cannot determine how effective a model is until we have built and tested it, this represents a considerable risk.

Signatures

Oliver Killane

Signed:

Date: 12/10/2022

Ethan Range

Signed:

Date: 12/10/2022

Bartlomeij Cieslar

Signed:

Date: 12/10/2022

Robbie Buxton

Signed:

Date: 12/10/2022

Charlie Lidbury

Signed:

Date: 12/10/2022

Jordan Hall

Signed:

Date: 12/10/2022

Dr Nicolas Wu

Signed:

Date: 12/10/2022