Development Plan
PCD: Partially Covered Detection of Obscured People using Point Cloud Data

Team #14, PCD
Tarnveer Takhtar
Matthew Bradbury
Harman Bassi
Kyen So

Table 1: Revision History

Date	**Developer(s)**	**Change**
September 24, 2024	Tarnveer Takhtar, Matthew Bradbury	Initial Draft

This report outlines the key aspects of our project, including our structured team meeting and communication plans. It details the roles and responsibilities of team members, our workflow plan, and our project decomposition and scheduling strategy. Additionally, it covers our proof of concept demonstration plan, the expected technology stack, and the coding standards we will adhere to. The appendices include reflections and our team charter, which highlights our external goals, attendance expectations, and accountability measures.

1 Confidential Information?

No confidential information present at this time.

2 IP to Protect

No IP to protect at this time.

3 Copyright License

Our project will be using the MIT License.

4 Team Meeting Plan

We will be meeting once a week, apart from lecture times, in a physical location on McMaster campus. Alternatively, the meeting can be held virtually on Discord or Microsoft Teams incase of unforeseen circumstances.

We will provide our advisor with updates bi-weekly, scheduling an meeting for larger updates and sending an email for smaller updates. Meetings will be held virtually unless for specific reasons. Virtual meetings will be hosted on Microsoft Teams.

For each meeting the scheduler of the meeting will be the chair. For recurring weekly meetings, the team lead will be the chair. Agenda will be prepared by the chair of the meeting.

The structure of the meetings will follow a simple outline. The meeting will begin with the discussion of last weeks meeting actions. It will then cover any new issues that have risen since our last meeting, followed by a discussion of what is need to be completed this week. The meeting will end with a quick summary of next weeks actions and how the work will be divided for the following week.

The advisor meetings will follow a similar structure, but are there to a dedicated time to answer the advisors questions and ask for help for any incompleted tasks. They will also end with a quick post of the meeting minutes from the discussion.

5 Team Communication Plan

Github issues, and a regularly updated Kanban board using those issues, will be used as a means of asynchronous communication between members and our Advisor outside of regularly scheduled meetings hosted on Microsoft Teams. Discord will be used for general communications between team members. Our Advisor will be contacted via Email or Microsoft Teams.

6 Team Member Roles

Team Lead:
Tarnveer Takhtar
This role entails communication with instructors/supervisors and keeping team members on task during meetings.

Approver:
Matthew Bradbury
This role entails reviewing the created pull requests as a final check before approving them for merge.

Reviewer(s):
Harman Bassi
Kyen So
This role entails reviewing documentation/coding changes prior to creating pull requests.

Administrator:
Kyen So
This role entails keeping notes on meeting points discussed and issuing action items to corresponding team members.

Coordinator:
Harman Bassi
This role entails planning out meeting schedules and ensuring no scheduling conflicts for team members.

All members will work together to be responsible with breaking down responsibilities in terms of deliverables. Furthermore, these roles are somewhat flexible based on the availability of each team member; some member may ask another

to cover their responsibilities if needed. Additionally, all members will be responsible for preparing the necessary background needed for this project. As mentioned in the problem statement, this is an advanced project partially due to the lack of experience each member has with computer vision. Therefore, it is the responsibility of each member to ensure that the necessary background is fulfilled.

7 Workflow Plan

We will be using a branch for each Revision, and separate branches for each major code features. Pull requests will be reserved for merging feature branches, while documentation changes will be manually reviewed and edited.

Github issues will be created for any tasks as per the course outline, issues will also be created for code features. Issue priority and classification will be designated by both deadline, and if it is a pre-requisite of other tasks. Issues will be assigned accordingly as discussed within our meetings. If any discrepancies are found within the documentation, an issue will be made, and assigned.

GitHub Actions will be used to handle CI/CD, triggered each time a pull request occurs for the codebase of the project. A predefined linter will go over the code ensuring consistency with the chosen linter's standard. Additional workflow tests may also be present to ensure compatibility.

8 Project Decomposition and Scheduling

Our Github project will consist of a Kanban board, highlighting the tasks that still need to be done, in progress or completed.

Link to Kanban Board: https://github.com/users/takhtart/projects/3

The Kanban board will contain a high-level overview of the tasks that need to be done, with priority based on deadlines provided by the course outline.

9 Proof of Concept Demonstration Plan

Significant Risks

1. Project Complexity: The primary risk is the potential complexity of the project, which may be greater than initially anticipated. Our team welcomes the challenges associated with this project however, we must understand the risks of taking on a project where we have no experience in the specific domain.

2. Kinect Sensor Performance: Another critical concern is the performance of the provided Kinect sensor, both during the development phase and the demonstration. Ensuring the sensor functions optimally and produces a quality real-time point cloud image is crucial for the project's success.

Overall, The performance and reliability of the Kinect sensor and the implementation complexity of the detecting people when only partially visible are our main concerns regarding this project.

Demonstration Plan

To demonstrate that these risks can be overcome, we plan on showcasing the system's ability to reliably identify people **not** obscured by any objects (base case). This will be a key achievement for our team, instilling confidence within our team that we can take on the primary challenge of identifying people when obscured. This process will also present the Kinect sensor's performance during both development and the demo, providing more first hand knowledge on the sensor's benefits and drawbacks we should be aware of.

10 Expected Technology

We plan to use Python as our primary programming language. Python is well-suited for data processing and machine learning, which we will likely need for our project. We anticipate utilizing libraries such as Open3D for handling point cloud data with a Kinect as our input. We may also use NumPy and SciPy for numerical computations, and Scikit-learn for machine learning tasks.

Additionally, we may require the use of deep learning frameworks like TensorFlow or PyTorch to train our own models given that it's unlikely there are any existing models for the problem we're trying to solve. To maintain code quality, we will employ tools like Flake8 for linting and Pytest for unit testing. For code coverage we plan use Coverage.py to ensure thorough testing. For CI, we will likely use GitHub Actions to automate testing and deployment. Collaboration and version control will be conducted through Git and GitHub. Github Projects will be used to manage opened issue tickets, and be our primary tracker for project progression. Performance monitoring tools such as Valgrind may be considered for optimizing critical code sections (optimizing real-time detection).

11 Coding Standard

The coding standard we will adopt will be C++ 20.

Appendix — Reflection

1. Why is it important to create a development plan prior to starting the project?

   It is important to create a development plan prior to starting the project as it allows most of the heavy lifting behind project planning to be done before any work has started. It allows for expectations and workflow to be clearly defined before any issues arise. It also creates a document that can be referenced at other times to avoid confusion.

2. In your opinion, what are the advantages and disadvantages of using CI/CD?

   Employing CI/CD allows for better issue tracking and rollbacks. Utilizing CI/CD gives the opportunity for teams to better track individual issues and commits, leading to increased awareness and visibility on workflow issues. It also allows for easier time rolling back to a previous version in case something goes wrong. Some disadvantages are with the conceptual depth and speed. Ensuring a specific workflow and constant PR reviews can slow things down as contributors have to make sure that they are following the workflow properly and have to wait for PR reviews (when necessary). Furthermore, it is more effort to set up, both in the codebase and conceptually. The process has to be talked through and understood by all team members.

3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?

   Our group mainly debated how to set up our GitHub workflows. Initially, we considered using individual branches for each issue, but we ultimately decided on a more streamlined approach with two revision branches and separate forks. This allows us to effectively manage pull requests for merging feature changes into the codebase. We reached this agreement after discussing the benefits of clarity and collaboration in our development process.

4. What went well while writing this deliverable?

   This deliverable allowed for the group to be able to discuss standard goals vs stretched goals. Everyone contributed their ideas and we were able to come to a clear conclusion when it came to the goals and problem statement of the project. It also allowed us all to see where the project is headed and how we should properly prepare ourselves so that we can achieve our goals.

5. What pain points did you experience during this deliverable, and how did you resolve them?

   The biggest pain point during this deliverable was being able to decide which goals were too ambitious and outside our design scope. Some goals such as the aspect of outlining the human in the environment were broken up. There was a deep discussion on how to properly decide the goals, but at the end the group got a better understanding of the project.

6. How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?

   Because our project is presented by a professor, the team already had a pretty clear understanding of the goals that needed to be achieved for our project to be considered a success. The only issue was trying to ensure that the goals can be properly broken down so that a goal that seemed a bit ambitious could be broken into something that seems doable. For example, the human detection is broken into the Minimal and viable product goal and then also extended into the stretch goal. This was an important discussion that the group had to ensure that we deemed our goals to be doable.

7. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

   Every member of the group needs to acquire knowledge on Computer Vision and understanding how pcd files operate. Overall these are big topics and so the basics should be acquired by every member, but some specifics would be broken down to different parts for each member. Tarnveer and Matthew would be assigned to understanding how to track people on screen and map them on the screen. Harman and Kyen would be working on reading in the pcd files and understanding the PCL. The PCL will explore on aspects of boxing the points on screen. Tarnveer would also be responsible for understanding the real time coding aspect in c++. Matthew would also be assigned to acquire knowledge on improving the human outline based on better data. Harman will be assigned to understanding how to cancel out noise from the data set. Kyen will have to understand how to find the person based off the pcd and understand how to properly read the files.

8. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

   One approach for acquiring the knowledge would be to use the provided documentation for the libraries (PCL and OpenCV). This would provide a good fundamental understanding for the important aspects of the project.

   Another approach would be to just watch videos on the specific topics and try to understand from there. This would be able to provide a more visual explanations for the topics.

   Tarnveer: use documentation and videos Matthew: use documentation and videos Kyen: use documentation Harman: use documentation and videos

9. What went well while writing this deliverable?

   The deliverable was straightforward. We had a rough idea of the main hazards within our project and tried to make sure that we covered the main scope. The document writing was split between all of us. The document is pretty straightforward and we as a group were able to talk over the different sections and divide up the work.

10. What pain points did you experience during this deliverable, and how did you resolve them?

    The biggest pain point was probably discussing what would be some assumptions we had to make, but we were able to come to an agreement by communicating our points of why or why not.

11. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?

    All the risks were mainly thought of before the deliverable. We knew that we needed to ensure that the offline file is the correct format and that the system needs to make sure it is working with a Kinect sensor and not something else.

    The privacy risk was something we thought of at the informal interview.

12. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?

    Could be some performance risks and making sure that the performance of the software meets the goals/requirements for the project. Another risk could be in terms of privacy. The application is capturing sensitive data and so its important on how the application handles this data.

13. What went well while writing this deliverable?

    This deliverable was relatively painless and straightforward. We had already started thinking about testing plans during our SRS deliverable, specifically section S.6. In this section we detailed a brief VnV plan, including system testing and unit testing. This set up the basic outline for this deliverable, it being an extension of what we already wrote/thought about.

14. What pain points did you experience during this deliverable, and how did you resolve them?

    One pain point we had during this deliverable was editing requirements from the SRS. When creating the traceability matrix, we noticed that some of the requirements from the SRS document were overlapping or in the wrong spot. We held a meeting as a group to sort this out and reach a consensus on which requirements should stay, should be changed, or should be deleted.

15. What knowledge and skills will the team collectively need to acquire to successfully complete the verification and validation of your project? Examples of possible knowledge and skills include dynamic testing knowledge, static testing knowledge, specific tool usage, Valgrind etc. You should look to identify at least one item for each team member.

    In order to properly complete the verification and validation of our project, some skills will need to Be acquired. Firstly, we will have to familiarize ourselves with cppunit, as majority of our testing knowledge from previous courses is in Java or Python. Additionally, because of this, we will have to learn how to use GCov in order to accurately figure out our code coverage. On the topic of code coverage, we will also need to brush up on our coverage definitions that were learnt in our testing course. Finally, we will need to implement linters to check our code on github before merge.

16. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

    For these skills, there are a few ways to approach acquiring the knowledge. With new skills, utilizing Youtube and online tutorials are a good way to quickly learn the basics and proper implementation of new techniques. For older skills, i.e. ones that we have learnt previously but haven't used in a while, we can go back to old projects/lectures and relearn the information.

    Matthew will find online tutorials to learn about cppunit. This is because he has no experience with creating automated testing in c++, and needs to start off by learning the basics.

    Tarnveer will find online tutorials to learn about cppunit and c++ linters on Github. This is for the same reason as above; he has no experience implementing testing in c++ or adding linters to Github for PRs.

Harman will go back to our old 3SO3 notes in order to relearn MC/DC coverage. This is because he had implemented MC/DC coverage and checks in that course, but in Java. Since he has implemented this before, he is familiar with the content and should be relatively painless to relearn the content.

Kyen will find online tutorials for learning about GCov. For similar reasons as Matthew and Tarnveer, this is because he has no prior experience with this specific coverage tool.

17. What went well while writing this deliverable?

    Everyone on the team was on track with their sections of the assignment and we were able to thick of better ways to break up some modules to make more sense.

18. What pain points did you experience during this deliverable, and how did you resolve them?

    Getting used to the new year and so it was a slow start trying to get back into the flow, but once we started working it came back.

19. Which of your design decisions stemmed from speaking to your client(s)or a proxy (e.g. your peers, stakeholders, potential users)? For those thatwere not, why, and where did they come from?

    Most of the module break up comes from talking to our client becuase they helped us focus on their vision for the project but making the inputs and specific variables were all done independently.

20. While creating the design doc, what parts of your other documents (e.g. requirements, hazard analysis, etc), it any, needed to be changed, and why?

    For now no real document had to be changed becuase the structure for this assignment was thought of before through the many client meets. This allowed for a strong structure.

21. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)

    With unlimited resources the ability to capture better imaging with the kinect would allow for a faster and more precise human detection algorithm. Maybe also being able to better maximize the human detection to better fit a humaniod shape.

22. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select the documented design? (LO_Explores)

    Other design implications would just involve taking a different approach to creating the algorithm. The issue with for example a solution that does

not use hue or skin color is limiting our ability to full captalize on the fact that the sensor picks up RGB as well.

Appendix — Team Charter

External Goals

The team's external goals for this project are to create an impressive project to discuss during interviews. Additionally, we aim to showcase a challenging and noteworthy project at the capstone convention. We also want to develop new skills and refine our existing ones throughout this project.

Attendance

Expectations

The expectation is that members clearly communicate their availability, and give at least 24 hour notice when not being able to attend meetings. If a member needs to skip/leave early, they are expected to communicate what work they will complete prior to the next meeting and ensure that they are up to date on any project changes. While there is no % of meetings that must be attended, we will be using group consensus to determine if one particular group member is missing a worrying number of meetings.

Acceptable Excuse

An acceptable excuse for missing a meeting or deadline would be any truly unforeseen/unavoidable engagement which requires immediate attention or any engagement that is properly communicated beforehand. The second type of excuse will not be permitted if the rest of the group feels that an individual member has missed too many meetings/deadlines.

In Case of Emergency

In case of emergency, the individual should convey exactly what they can and cannot accomplish to the team, and ensure that they complete an adequate amount of work at a later date for whoever ends up covering.

Accountability and Teamwork

Quality

Team members are expected to adequately prepare for team meetings. This means completing agreed upon work prior to meeting times, barring a reasonable excuse. For supervisor/TA meetings, members are expected to adequately understand the reason for the meeting enough to contribute unique ideas/questions to the conversation.

Regarding quality of work, members are expected to complete work at a reasonable pace, allowing room for adequate review to take place prior to deadlines. The work should also be done at a quality similar to the others, i.e. everyone

is contributing at a quality similar to the person of highest engagement/quality level.

Attitude

The team's expectations are that each member completes work with a level of engagement relatively similar to other team members. When providing ideas/interactions, it is expected that each member is treated with respect.

Stay on Track

We will use Github projects to ensure that we are on track. Each task to work on will be stored in an issue, which will have up to date deadlines attached to them, as well as the assignee(s). This will give us the ability to check up and see if tasks are being worked on/reviewed at a timely pace. On top of issues, we will be looking at commits and team meeting attendance to gauge contributions.

In the event that members aren't contributing their fair share, that member will have to purchase snacks for the next team meeting. We will also have a group meeting with them to discuss steps to improve and expectations. If the behaviour continues further, we will escalate the issue to a TA/professor.

Team Building

Team cohesion will be built through friendly conversation/banter and engaging in fun/relaxing group activities.

Decision Making

We will make decisions based on group consensus, and if the vote isn't unanimous we will try to solve the dispute via discussion until the decision is unanimous. If disagrees propagate further, we will escalate to asking our supervisor or TA for advice on how to proceed.