Hazard Analysis
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**
October 25, 2024	T. Takhtar, M. Bradbury, H. Bassi, K. So	Initial Draft

Contents

1 Introduction

This document contains details of our Hazard Analysis on our Partially Covered Detection software system. A hazard will be defined as a condition or situation where privacy, security, and/or system reliability may be compromised.

2 Scope and Purpose of Hazard Analysis

The purpose of this Hazard Analysis is to identify, evaluate, and mitigate potential risks associated with the Partial Covered Detection software system. By identifying the associated hazards, we can further ensure the privacy and security of users and their data, and increase the reliability of the system.

The scope of this analysis includes the entire lifecycle of the detection system. This entails the planning, development, deployment, and maintenance phases of the software cycle.

Potential Losses Due to Hazards

1. **Privacy Violations:** Unauthorized capture and storage of point cloud data by the Kinect sensor, without permission from parties present in the capture could lead to breaches of user privacy. This may lead to potential legal repercussions due to non-compliance with data protection regulations.
2. **Legal and Regulatory Consequences:** Non-compliance with data protection laws and regulations may incur fines and legal actions.
3. **Operational Disruptions:** Any failure in system performance, such as incorrect data inputs/outputs or unresponsiveness could disrupt critical operations, leading to reduced functionality in real-time applications.

3 System Boundaries and Components

The application is divided into three components

3.1 Human Detection System

The application can read in data and provide the location of the human. The system will provide the information needed to map the head, torso, and/or limbs.

3.2 Human Outline System

The application maps out the outline of the human within the environment.

3.3 File Manager System

The application takes in information from either the offline file or the kinect sensor and filters out the noise from the files.

4 Critical Assumptions

- **Kinect is function correctly and provides correct,usable outputs:** Since the hardware of the kinect and its outputs are out of the scope of the project, we assume that the kinect is functioning correctly and is producing correct and usable outputs for our software to use.
- **Kinect is placed in the environment it is designed for:** Our software will fail to function properly if it is given data collected by a kinect that is placed in an environment our software isn't designed for. This includes the outdoors or a room with insufficient lighting or without sufficient lighting.

5 Failure Mode and Effect Analysis

Component	Failure Mode	Effects of Failure	Causes of Failure	Recommended Action	Safety and Functional Requirement	Ref
Human Detection Algorithm	Human not detected	Lack of Reliability, software output not accepted	Software is unable to distinguish a human from their background and the objects that partially cover them due to a computational error	Update algorithm and tweak it to work against various body types	[F221]	4.1.1.0
	Incorrect object detected as human	Lack of Reliability, software output not accepted	Software incorrectly identifies an object to be the body part of a human due to a computational error	Update algorithm and ensure software is able to distinguish between common objects and human body parts	[F221]	4.1.1.1
Human Outline Manager	Non-human object is outlined	Lack of Reliability, software output not accepted	Software incorrectly identifies an object to be the body part of a human and outlines the object to display to the user due to a computational error	Update algorithm and ensure software is able to distinguish between common objects and human body parts	[F211][F222][F223]	4.1.2.0
	Outline missing for parts of human	Lack of Reliability, software output not accepted	Software is unable to identify certain parts of a human that the Kinect captures and thus, fails to outline those body parts due to a computational error	Update algorithm and tweak it to work against various body types	[F211][F222][F223]	4.1.2.1
	Human is outlined incorrectly	Lack of Reliability, software output not accepted	Software is unable to accurately produce an outline of the human captured due to a computational error	Update algorithm and tweak it to work against various body types	[F211]	4.1.2.2
Kinect Manager	Kinect connection disconnects	Catastrophic failure for live detection, Software immediately stops and ceases to produce any outputs	Loose wire causing the Kinect to disconnect Accidental disconnection of the wire	Error output to the user, system pauses and prompts user to check the connection of the Kinect to the computer	[F231][F232]	4.1.3.0
	Software fails to read data from Kinect	Catastrophic failure for live detection; software is unable to perform its functions and produce any outputs	Data corruption, incorrect Kinect model used, incorrect resolution received	Error output to the user, system pauses and prompts user to check if they are using the correct version of Kinect	[F231][F232]	4.1.3.1
	Unsupported file uploaded	Software is unable to perform its functions	User uploads a file with data that is not supported by the software	Error output to the user, system pauses and prompts user to reupload a .pcd file that was collected by a Kinect	[F231][F232][SR4]	4.1.3.2

Table 2: Failure Mode and Effect Analysis

6 Safety and Security Requirements

[SR1] The application must encrypt user entered files.
Rationale: This is to ensure the privacy of the files uploaded by the user

[SR2] The application must not store any files uploaded.
Rationale: This is to ensure that in case of any data breach the files will not be something an attacker could have access to.

[SR3] The application must not store the .pcd files sent from the live Kinect.
Rationale: To ensure that the live sensor data protects the user's privacy within the environment.

[SR4] The application must reject any foreign or unexpected file formats.
Rationale: This is to ensure that our program will not have any unexpected behaviour following improper use.

7 Roadmap

In terms of the the roadmap, we will be implementing some of the newly discovered requirements while leaving others for future plans. During this capstone, we will ensure that files will not be stored and that unexpected file formats will not be allowed to be uploaded. In the future, past the scope of this capstone, we would implement some form of data encryption on all files that go into our system, as well as output files generated.

To summarize, the requirements we want to include in the scope of this capstone, we will be implementing prior to the end of the course. Which is to say that these requirements will be implemented by April 2, with the rest of the source code. The other requirement(s) will not be implemented within the course of this capstone.

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.