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
March 27, 2025	Harman Bassi	Revision 1
March 27, 2025	Harman Bassi	Revision 1
March 27, 2025	Harman Bassi	Revision 1
March 27, 2025	Harman Bassi	Revision 1
March 27, 2025	Harman Bassi	Revision 1

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][F222]	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][F222]	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][SR3]	4.1.3.2
	Uploaded file is stored	Lack of Privacy and Data Security, software could possible be leaked	User uploads a file and it was saved by the software	After data has been uploaded and the user is done with this data set, system will reset the stored data value to the starting point.	[SR1][SR2]	4.1.3.2

Table 2: Failure Mode and Effect Analysis

The Functional and Non-Functional Requirements in the table can be found in the SRS document

6 Safety and Security Requirements

[SR1] 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.

[SR2] 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.

[SR3] 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. 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.

2. 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.

3. 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.

4. 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.