Partially Covered Detection of Humans

Requirements Standard Plan

Tarnveer Takhtar, Matthew Bradbury, Harman Bassi, Kyen So

Version 0, October 11, 2024

Table of Contents

Academic Integrity Disclaimer

We would like to acknowledge that as a dedicated students of McMaster University, we have thoroughly read and comprehended the Academic Integrity Policy published by the university. We are committed to upholding the principles of academic honesty and integrity in all aspects of our educational journey. We understand the importance of acknowledging the work and ideas of others, and we pledge to ensure that all our academic endeavors are conducted with the utmost originality and compliance with the university's policy.

We affirm that the content presented in this document is entirely our own, and any external sources used have been appropriately cited and referenced.

Tarnveer Takhtar

As I submit my work, I, **Tarnveer Takhtar**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Matthew Bradbury

As I submit my work, I, **Matthew Bradbury**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Harman Bassi

As I submit my work, I, **Harman Bassi**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Kyen So

As I submit my work, I, **Kyen So**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Control Information

Version	Delivery		Updates	
	Deadline	*Delivered*	*Change*	*Date*
V0	Oct 11, 2024	Oct 11, 2024	Initial SRS Draft	Oct 11, 2024
Rev 1	Apr 2, 2025	TBA	Revision 1	March 26, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 26, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 26, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 26, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 26, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 26, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 26, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 27, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 27, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 27, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 26, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	March 26, 2025
Rev 1	Apr 2, 2025	TBA	Revision 1	April 2, 2025

(G) Goals

(G.1) Context and Overall Objectives

Currently within robotics at McMaster, pushing forward the use of robots with computer vision is a main objective. With the current technology, there is apt ability for the robots to detect obstacles and avoid them. With this project, we aim to fill the aforementioned gap and provide an algorithm to equip these robots with the ability to detect people, with a specific focus on partially detected people, in real time. This project will be used by graduate students and professors in the Mechanical Engineering lab for future research projects.

Specifically, our goals for this project are to detect partially detected people in real time. This can be broken up into a few sub-goals as follows. First, we need to ensure that we can detect people. This is imperative for our our POC demo, in which we want to be able to properly detect people outside of cover. After this, we need to map a person, and their body parts, to shapes. This lays down the groundwork for the ability to detect partially covered people, through figuring out which body part is showing. We also need to ensure that our output includes location data, i.e. the precise coordinates of the detected human. Additionally, we need to ensure that we are able to process and produce these outputs in real-time. Finally, along with the ability to process point cloud data live from a Kinect, we need to be able to process raw .pcd files offline.

(G.2) Current situation

The current state of robotics research, specifically at McMaster, does not include the ability to process Kinect sensor data, or .pcd file, and detect what is a human versus an obstacle on screen. Regarding robots using computer vision sensors, the current situation is that they have the ability to detect obstacles as well as the ability to stop moving upon detecting an obstacle in the robot's path. As mentioned in the Overall Objectives section, this creates a gap where we are unable to differentiate humans from obstacles, as humans are currently considered the same as an obstacle: something that is blocking the robot's path. In order to proceed with more robust options for research with these robots, especially regarding interactions between robots and humans, there must be some way for the robot to detect humans apart from random obstacles for safety concerns or to perform certain actions. Additionally, the ability to take in PCD data and detect not just humans, but specific body parts, the research applications that may spawn from this widen significantly.

(G.3) Expected Benefits

From successful execution of the project, we hope to achieve the following benefits:

Enhanced Human-Robot Interaction: By developing software that can reliably detect partially hidden people, the project will improve the safety and effectiveness of human-robot interactions by increasing the awareness capabilities of the robot.

Real-Time and Offline Processing: The ability to process data in real-time using a Kinect sensor, as well as offline using .pcd files, will provide flexibility for the user to view detection live, or view detection results using previously captured data. This ensures that the system can be tested and used in different scenarios, enhancing its robustness and reliability.

Improved Skeleton Tracking: The project aims to overcome the limitations of the existing skeleton tracking offered by the Kinect SDK by providing more advanced estimations of a person's body, even when partially obstructed. This improvement will be beneficial for robotics applications.

Versatility in Body Poses and Scenarios: The software will be tested against a variety of cases. For example a person in frame may not ne facing the sensor directly, could be in a different position, or be partially obstructed by objects or other individuals. Being able to accommodate in these situations will make the system more adaptable to real life situations, and have more practical value.

Contribution to Research and Development: The project will contribute to the broader field of robotics and computer vision by addressing a challenging problem with a real world application and provide a potential solution. This can pave the way for further research and development in assistive robotics and related areas in the future.

(G.4) Functionality overview

(G.4.1) Functional Requirements

1. **Human Detection:** The system must detect humans within the sensor's field of view, even if they are partially obscured by objects or other people. (F411) Satisfies [NF431], [NF432]
2. **Location Estimation:** The system must be able to estimate the current space occupied by the detected human and provide a visual representation of that space estimation. (F412) Satisfies [NF433]
3. **Offline Processing:** The system must be able to analyze uploaded .pcd files for offline testing and validation. (F413) Satisfies [NF431]
4. **Body Pose Variation Handling:** The system must handle a variety of body poses and orientations, and not be limited to just the human facing the sensor. (F414) Satisfies [NF432]
5. **Integration with Kinect Sensor:** The system must integrate seamlessly with the Kinect sensor for data collection. (F415)
6. **Real-Time Processing:** The system must process data in real-time with minimal latency to ensure timely detection and response in a live detection setting. (F416)

(G.4.2) Top Two Most Important Functional Requirements From A Business Perspective

The ability to reliably detect and model humans even when partially obscured is vital for the system's success, as it directly impacts the accuracy and reliability of human-robot interactions in real-world scenarios. 3D Space Estimation will also ensure that the system can accurately identify and understand

the spatial context of the area, allowing for more precise interactions and movement planning by assistive robots.

(G.4.3) Non-Functional Requirements

1. **Usability:** The system must provide an intuitive and easy to use interface. (NF431) Depends on [F411] ,[F413]
2. **Reliability:** The system must provide consistent and accurate detection and modeling of humans. (NF432) Depends on [F411], [F414]
3. **Accuracy:** The system must provide an accurate visualization the space occupied by detected humans. (NF433) Depends on [F412]

Non-Functional Requirements Justification

Real-time processing is essential for practical applications, ensuring timely detection and response. The system must be adaptable to different environments and varying numbers of people to be useful in diverse scenarios. Consistent and accurate detection is crucial for the system's effectiveness and user trust. Accurate detection and modeling are critical for the system's success and practical applications, ensuring that the system provides reliable and precise information for human-robot interactions.

(G.4.4) Traceability Matrix

The following goals can be found in Problem Statement

Requirements	Goals	Connected Requirements
[F411]	2.1	Satisfies [NF431], [NF432]
[F412]	2.3	Satisfies [NF432]
[F413]	2.5	Satisfies [NF431]
[F414]	2.1	Satisfies [NF433]
[F415]	2.2	-
[F416]	2.2	-
[NF431]	2.4	Depends on [F411], [F413]
[NF432]	2.1	Depends on [F411], [F414]
[NF433]	2.3	Depends on [F412]

(G.5) High-level usage scenarios

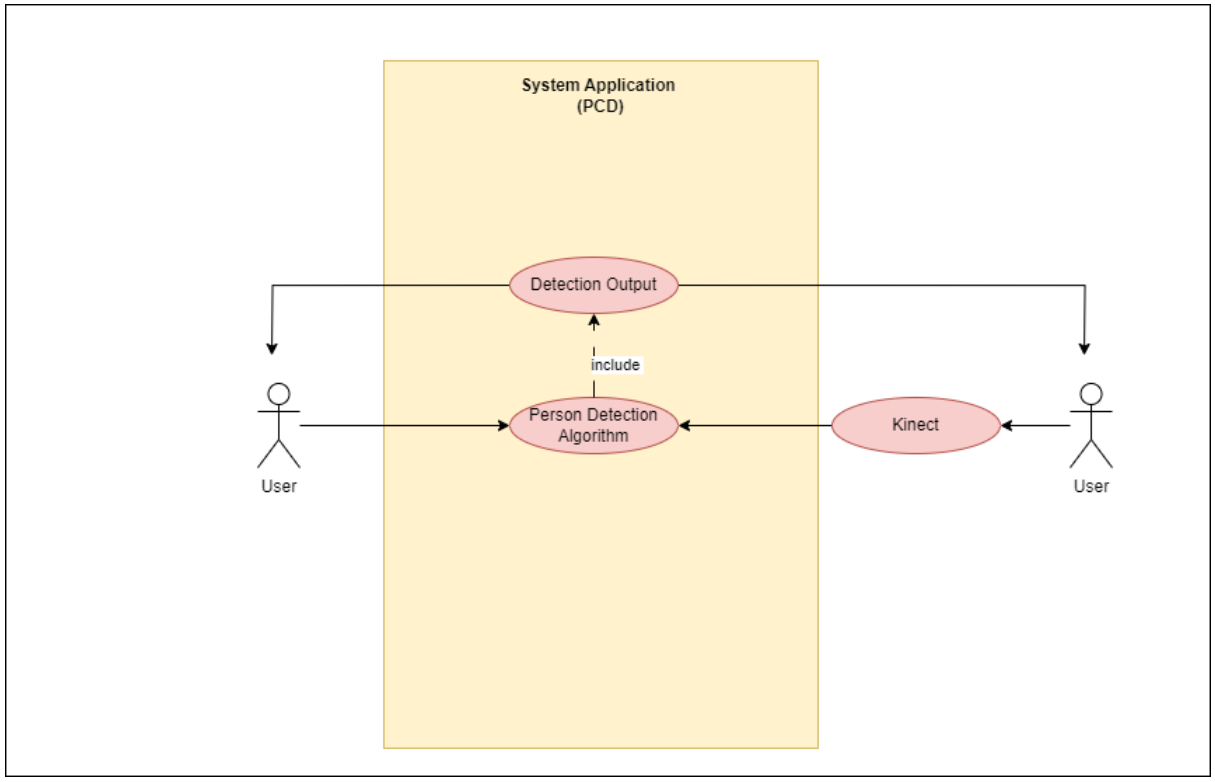


Figure 1. High Level use cases diagram

- **Connect Kinect Sensor for Live Feedback**
  - User begins detection software on device.
  - User connects Kinect to device running detection software.
  - Detection software takes the .pcd from the Kinect.
  - Detection software performs calculations.
  - User observes output, a copy of the live .pcd feed with coordinates of the person and a shape mapped onto the visible body parts. Satisfies [F411], [F412], [F414], [F416]
- **Import Raw .pcd file**
  - User uploads .pcd files to device.
  - User begins detection software in offline mode, specifying desired .pcd files.
  - Detection software runs on the specified .pcd.
  - User observes output, a copy of the uploaded .pcd file with coordinates of the person and a shape mapped onto the visible body parts. Satisfies [F411], [F412], [F414], [F416]

(G.6) Limitations and Exclusions

Along with the exclusions mentioned in E.3, the project will not be focusing on the following aspects of the detection system:

Fully Obscured Handling: The system will not address scenarios where a person is completely occluded by objects or other individuals. The focus will be on detecting and estimating the 3D space occupied by partially visible people.

Non-Human Object Detection: The system is specifically designed to detect and estimate the space occupied by humans. It will not be responsible for identifying or tracking non-human objects or animals.

Multiple Sensor Integration: The project will utilize a single Kinect sensor for data collection. Integration of multiple sensors or different types of sensors is beyond the scope of this project.

Advanced Human Pose Estimation: While the system will be able to handle a variety of body orientations, it will not focus on advanced human pose estimation techniques beyond the basic extrapolation of visible parts of the body to detect a person.

Long-Term Data Storage and Analysis: The system will process data in real-time and offline using .pcd files, but it will not include long-term data storage or analysis capabilities. Data management and storage solutions are outside the scope of this project.

User Interface Development: The primary focus of the project is on the development of the detection and estimation software. The creation of a user-friendly interface for end-users beyond providing a depot of uploading a .pcd file, is not a primary focus of the project.

(G.7) Stakeholders and requirements sources

Stakeholders

Academic Researchers: This category includes professors, researchers, and students in the fields of robotics, computer vision, and artificial intelligence. Their input is crucial for ensuring the project aligns with current research trends and academic standards.

Requirements Sources

Academic Publications: Research papers, journals, in the field of computer vision, and possibly on artificial intelligence. These documents provide valuable insights into current methodologies and technologies.

User Feedback: Surveys, interviews, and usability studies conducted with end-users. This feedback helps identify user needs and preferences.

Technical Documentation: Manuals, specifications, and other technical documents related to the Kinect sensor and the Kinect SDK, Point Cloud Library (PCL), and other tools used in the project.

(E) Environment

(E.1) Glossary

.pcd

Point Cloud Data, the file format that our code will be receiving and performing all computer vision calculations on.

Kinect

The Kinect refers to an Xbox One Kinect Sensor, which we will be interacting with and using to gather .pcd files.

Kinect SDK

The Kinect Software Development Kit (SDK) enables us to develop and deploy code/applications on the aforementioned Kinect.

P.C.D.

Partially Covered Detection, refers to our project title. Our project is centered around automatically detecting partially covered people using .pcd files.

OpenCV:

The Open Source Computer Vision library is an open source computer vision and machine learning software library. It will be used in our C++ detection algorithm to aid with image processing.

PCL:

The Point Cloud Library is an open-source library of algorithms for point cloud processing tasks. This library will be used in our code to aid the processing of .pcd files.

Person/Human

Any human subject within the field of view of the Kinect sensor or within the .pcd file.

(E.2) Components

The system will interact with the following external component(s)

Kinect Sensor: This component is responsible for delivering the system live .pcd data for the detection algorithm to run on.

Detection Algorithm This component will be used to handle all of the image processing related to being

able to detect a person.

.pcd Processor This component will be used in our code to aid the processing of .pcd files, and further help the detection algorithm in person detection.

(E.3) Constraints

- **Location of Kinect sensor**: If the Kinect sensor was placed outdoors, it could introduce unstable lighting and different weather conditions can prevent the microsoft Kinect from capturing reliable data. Making the detection software work in outdoor environments can pose additionally challenges which is unnecessary as the software is meant for indoor assistive robots. Therefore, the scope of our project will be limited to exclusively indoor use.
- **Lighting Levels**: If the Kinect sensor is placed in a room that is too dark or too bright, it can prevent the detection software from functioning correctly. Modifying the detection software to detect in the dark without color data proposes additional challenges beyond the scope of the project. Therefore, the Kinect must be placed in a room with ample but not excessive lighting unless the development team has enough time during the development period to work on the stretch goal that is detecting humans in the dark.

(E.4) Assumptions

- **Kinect Sensor compatibility**: It is assumed that the Kinect sensor is sending data in real time and that the data is reliable. This is to ensure that the sensor is sending data that can actually be used for the algorithm.
- **Standard Human Size**: The general size for a human is going to be assumed, so that there could be a starting point for the application. For the system to estimate where to person is hidden and their general size the algorithm will assume a general human size as default.
- **Library Reliability**: Assuming that the libraries used provide the correct information. Some libraries like OpenCV and the PCL will be used a lot and so its important to assume that the libraries do not have any errors that would affect the application.
- **Consistent Lighting Conditions**: The system relies on consistent lighting conditions to accurately detect and model humans.

(E.5) Effects

Some potential effects of this system are as follows:

- Increase in research opportunities. This is possible due to our system providing our stakeholders with additional options and functionality for robots using Kinect sensors, or other sensors that are able to produce .pcd files. The added functionality of being able to detect people, even when they are partially hidden, will allow researchers to experiment with many more scenarios and situations.

(E.6) Invariants

Sensor Calibration: The Kinect sensor must be properly calibrated and positioned to ensure accurate data collection. The Kinect sensor will remain in the same place once detection has started.

Unobstructed Sensor View: The sensor's field of view should be unobstructed during the system's operation. While the system itself will be designed to detect partially obscured people, the sensor should maintain it's full field of view and be unobstructed.

(S) System

(S.1) Components

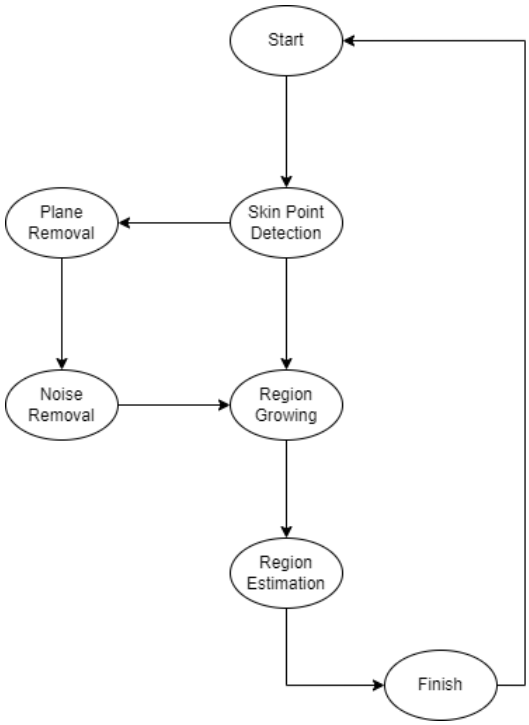


Figure 2. Detection Algorithm State Machine

The application is composed of 2 software components:

- **Human Detection Algorithm**: This component reads in the data from the Kinect manager and provides the location of the human. Based on the provided data it will present the precise location of the hidden person within the environment. The diagram above displays the logic for this detection algorithm. It will loop indefinitely as long as .pcd files are being inputted, either offline or from the Kinect. First the algorithm will attempt to find skin points within the point cloud. Then, it will undergo plane removal and noise removal while sending the original skin points to region growing. Points discovered within the planes/noise will be removed before region growing. After the region of the person is grown, the algorithm will estimate the space occupied by this detected person. Finally, the algorithm will finish and loop back around to start detection on the next frame/file.
- **Human Outline Manager**: This component is responsible for showcasing the Point Cloud Data alongside the human outline. The display will visualize the human within the 3D environment and provide a rough estimate of the person's location on screen.

The application is composed of 1 hardware component:

- **Kinect Manager**: The Kinect is the main hardware component for the P.C.D application. The Kinect Manager reads the provided Point Cloud Data of the environment. It will then filter out the noise from the data set to provide useful data. Finally sending this data to the algorithm to be processed. The

Kinect manager will constantly be filtering and sending data in real time.

(S.2) Functionality

(S.2.1) Human Outline Display:

Functional Requirements:

1. **Display the human outline**: The application displays the real time data as a 3D point cloud and showcases the location of the human within the data set. (F211)

Non-Functional Requirements:

1. **Appearance/Style of Display**: The application must have a design that is minimalistic. (NF211) **Rational**: The application should only display necessary items on screen such as the point cloud data and the human outline. The minimalist design will allow for easily readable data.
2. **Speed and Latency of Display**: The application must update the location of the human within __ milliseconds. (NF212) **Rational**: The expectations of a live display is that the application presents all information seamlessly and without any noticeable delay.

(S.2.2) Human Detection Algorithm:

Functional Requirements:

1. **Locate Visible Human Parts**: The application must use the data set provided and calculate the location of the visible parts of the person in the environment. (F221)
2. **Estimate Hidden Human Parts**: The algorithm must use the location of visible human parts to make a general estimation of the hidden human parts.

Non-Functional Requirements:

1. **Speed and Latency**: The application must process the data and find the human in every other frame. (NF221) **Rational**: Human detection must be processed within real time.
2. **Accuracy**: The application must be able to detect humans with 85% accuracy. (NF222)
3. **Reliability**: The application performs with a success rate of 90%. (NF223) **Rational**: The application is able to operate will a low failure rate so that the data presented is accurate

(S.2.3) Kinect Manager:

Functional Requirements:

1. **Read the Point Cloud Data**: The Kinect manager should be able to process the noise within the data to present a readable data set for the application. (F231)

2. **Send Data back to Algorithm**: The managed data is then sent to be computed by the detection component. (F232)

Non-Functional Requirements:

1. **Speed and Latency**: The Kinect manager is able to send back the RGB and depth data in 60-80 ms. (NF231)
2. **Accuracy**: The Kinect must be able to correctly cut out the noise effectively by 85%. (NF232)

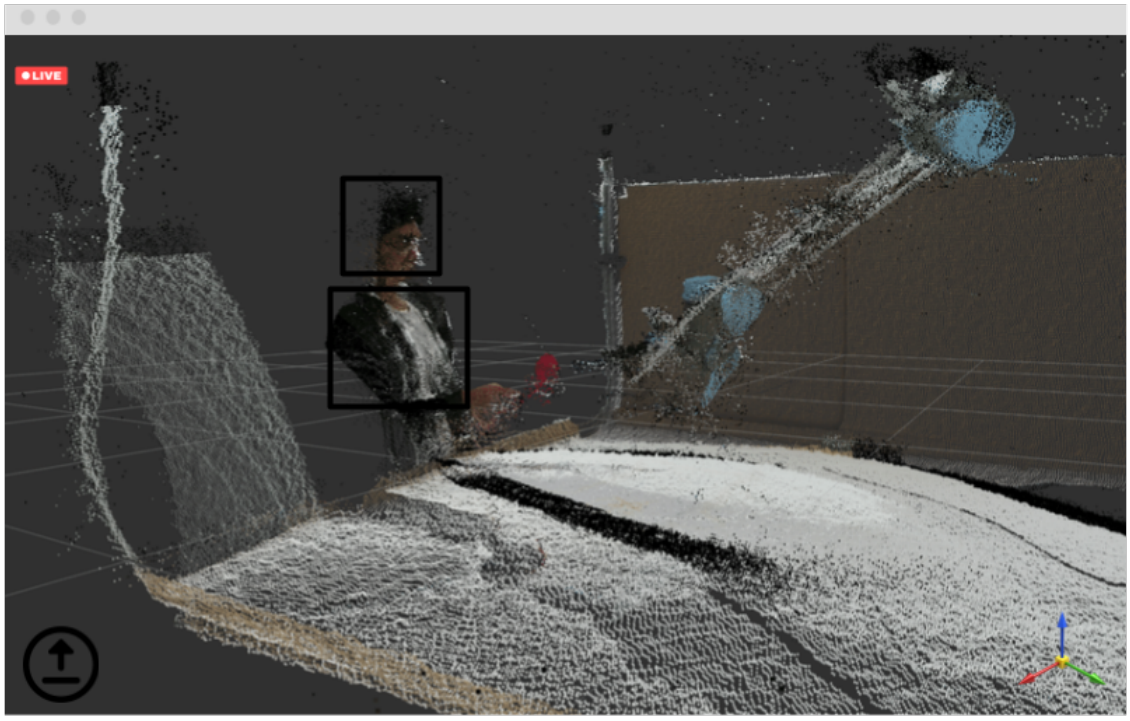
(S.3) Interfaces

(S.3.1) API

One of the main API for the application is the Kinect connection. The software takes in the real time data from the Kinect sensor and then processes that information within the algorithm. The Kinect sends this data to help the application find out where the person is hidden.

(S.3.2) Wireframe Mock-ups

Display Interface:



The main aspect of the display is the live or offline output that shows the Point Cloud Data with the outline of the person as certain shapes. The button on the bottom allows for the switching between live and offline display. The button allows the user to plug in the Point Cloud Data file.

(S.4) Detailed usage scenarios

(S.4.1) User starts realtime detection using the software

- **Use Case**: UC1
- **Primary Actor**: Graduate Research Student
- **Precondition**: Graduate Research Student has access to a Kinect, the detection software, a computer that can be used to connect to the Kinect and the wires necessary.
- **Trigger**: Graduate Research Student wishes to use the detection software to aid in their research
- **Main Success Scenario:**
    - 1. Graduate Student connects a Kinect to their computer using a wire
    - 2. Graduate Student opens the detection software

    - 3. Graduate Student configures the Kinect sensor and chooses "realtime detection"
    - 4. Software displays the video output of the Kinect sensor and visually outlines the humans it detects
    - 5. Graduate Student analyzes the output of the software to aid in their research
    - 6. Graduate Student finishes their work session
- **Secondary Scenario:**
    - 5.1 Graduate Student takes the output of the detection software and uses it as the input of their own project
    - 6.1 Graduate Student modifies and integrates the detection software to their project
- **Success Postcondition:** Graduate Student is provided accurate and reliable outputs to work with

This sceneario is important as it demonstrates the basic case of how a Graduate Student would interact with the software and use the software to aid in their research

(S.4.2) User uses software to analyze offline .pcd file

- **Use Case**: UC2
- **Primary Actor**: Graduate Research Student
- **Precondition**: Graduate Research Student has access to .pcd files they would like to use, the detection software, a computer to run the software.
- **Trigger**: Graduate Research Student wishes to use the detection software to aid in their research
- **Main Success Scenario:**
    - 1. Graduate Student opens the detection software
    - 2. Graduate Student chooses "offline file upload"
    - 3. Graduate Student uploads an offline .pcd file to the software
    - 4. Software provides the coordinates and distance of the humans it detects
    - 5. Graduate Student analyzes the output of the software to aid in their research
    - 6. Graduate Student finishes their work session
- **Secondary Scenario:**
    - 5.1 Graduate Studen takes the output of the detection software and uses it as the input of their own project
    - 6.1 Graduate Student modifies and integrates the detection software to their project
- **Success Postcondition:** Graduate Student is provided accurate and reliable outputs to work with

This sceneario is important as it demonstrates the other usage of the software which is another way a Graduate Student can interact with the software and use the software to aid in their research

(S.5) Prioritization

Priority Table for Human Outline Manager

Requirement ID	Requirement	MoSCoW Classification	Summary
[F211]	Display the human outline	Must have	It is important to be able to show on the screen where the person is hidden.
[NF212]	Speed and Latency	Must have	It must be able to present all the environmental and camera changes in real time.
[NF211]	Appearance/Style	Could have	It could have a minimalistic design to make reading the data easier. It is not necessary, but would make the output more clear.

Priority Table for Human Detection Algorithm

Requirement ID	Requirement	MoSCoW Classification	Summary
[F221]	Locate Visible Human Parts	Must have	It is important that the algorithm is able to locate where the visible part of the human is.
[F222]	Estimate Hidden Human Parts	Must have	It is important that the algorithm is able to take the visible parts and then estimate where it believes the rest of the human would be.
[NF221]	Speed and Latency	Must have	It must be able to calculate everything in real time to present the newest location and outline.
[NF222]	Accuracy	Must have	It must give an accurate location and outline.

[NF223]	Reliability	Must have	It must have an algorithm that has a low failure rate and make sure that the latest, most accurate data is shown.

Priority Table for Kinect Manager

Requirement ID	Requirement	MoSCoW Classification	Summary
[F231]	Read the Point Cloud Data	Must have	It is important that the manager is able to properly read in the data from the Kinect sensor and combine the two data sets together.
[F232]	Send Data back to Algorithm	Must have	It is important that the data sent back to the algorithm accounts for the noise in the environment and that the data being sent is the latest.
[NF231]	Speed and Latency	Must have	It must regulate the noise and send the data as fast as possible so that the application is in real time.
[NF232]	Accuracy	Must have	It must give back accurate data so that the application does not fail.

(S.6) Verification and acceptance criteria

For the functional requirements:

- The form of testing that will be done is unit testing, integration testing, acceptance, and system testing for validation. The Kinect sensor compatibility will be tested in the earlier stages of development as it is the most crucial part of the application. Integration testing will be done with the Kinect to ensure the sensor is properly connected to the application. The unit test will be done to ensure that every relevant step within the application outputs correct answers. Multiple different system tests for the application will be done by setting up the application in a variety of environments. Finally, relevant stakeholders can check that all functional requirements are met by the application before it is ready

for a final release.

For the non-functional requirements:

- It is important to conduct performance testing and accuracy testing. The system needs to perform in real time meaning the performance must be tested by the time outputs of the application. Testing the application in multiple environments and checking for latency will allow for a variety of performance tests to see the capabilities of the application. Accuracy testing would be done to see if the output on the display matches the correct state of the environment. For example, if a person's hand is seen as the only thing in the environment from the sensor's perspective.

For Usage Scenario S.4.1:

We have identified S.4.1 to be the most important scenario because it represents the main functionality of the software. Using Gherkin scenarios, we can outline a few tests to validate this scenario:

- **1. Scenario:** Graduate Student analyzes the output of the software to aid in their research
  - Given: Graduate student uses the realtime detection feature of the software for their research
  - When: Graduate Student finishes configuring the Kinect sensor and starts the live detection
  - Then: Detection software displays and outlines the humans it detects
  - And: Detection software provides the coordinates and distance to the humans it detects
  - And: Graduate Student uses the output of the software to aid in the development of their project
  - Then: Graduate Student finishes their work session
- **2. Scenario:** Graduate Student takes the output of the detection software and uses it as the input of their own software
  - Given: Graduate student uses the realtime detection feature of the software for their research
  - And: Graduate student has their own software ready that can work in conjunction with the detection software
  - When: Graduate Student finishes configuring the Kinect sensor and starts the live detection
  - Then: Detection software displays and outlines the humans it detects
  - And: Detection software provides the coordinates and distance to the humans it detects
  - And: Graduate Student feeds the output of the detection software to the input of their robotics software
  - Then: Graduate Student finishes their work session

(P) Project

(P.1) Roles and personnel

- **Supervisor**: The supervisor of the project will provide guidance and outline the requirements of the project. The supervisor is responsible for meeting with the software development team to answer questions and provide input on design decisions.
- **Software Developers**: The software developers are responsible for designing and implementing the algorithm used to identify partially covered humans and outlining body parts for the user to see. They are also responsible for testing and ensuring the accuracy of the software.

(P.2) Imposed technical choices

- **Realtime Computation**: Software must be able to perform calculations and output/display results in realtime.
- **Data gathered by Kinect sensor**: Software must be able to analyze data gathered using the Microsoft Kinect Sensor. Test data collected by a Microsoft Kinect, as well as the hardware itself, is provided for the project.

(P.3) Schedule and milestones

Key Dates:

- **September 23, 2024** : Problem Statement, POC Plan, Development Plan
- **October 11, 2024** : Revision 0 of Requirements Document
- **October 23, 2024** : Hazard Analysis
- **November 1, 2024** : V&V Plan
- **November 18, 2024** : Proof of Concept Demonstration
- **January 15, 2025** : Revision 0 Design Document
- **February 3, 2025** : Revision 0 Demonstration
- **March 7, 2025** : Revision 0 V&V Report
- **March 30, 2025** : Revision 1 Final Demonstration
- **April 8, 2025** : EXPO Demonstration
- **April 2, 2025** : Revision 1 Final Documentation

(P.4) Tasks and deliverables

Date	Tasks and Deliverables	Main Activities	Results
September 23, 2024	Problem Statement, POC Plan, Development Plan	Defined the initial problem statement, POC Plan and Development Plan	An initial problem statement, POC Plan and Development plan that aligns with Dr. Bones' needs and requirements
October 11, 2024	Revision 0 of Requirements Document	First Revision of our requirements document using the Meyer's Handbook as a template. This includes all four PEGS of requirements engineering. The Project section outlines features of the development project. The Environment section covers properties the project must abide by but cannot control such as constraints from physical law, business rules or engineering constraints. The Goal section covers business benefits we are trying to achieve with our project. Finally, the System section covers the performance and behaviour of the system.	Complete initial draft of our requirements based on our current assumptions and expectations
October 23, 2024	Hazard Analysis	Using various hazard analysis techniques to ensure that the project meets all safety requirements	Analysis of various possible system fail states and mitigation plans are formed

November 1, 2024	V&V Plan	Detailed testing plan and timeline for each module of our system to ensure that our project meets the software requirement specifications	A complete testing plan that we would follow to ensure that our code is accurate and reliable
November 18, 2024	Proof of Concept Demonstration	Delivery of a MVP to demonstrate the viability of our project to our stakeholders. Our team will also try to change our challenge difficulty to advanced.	Successfully demonstrate a working MVP so we could move on with the project
January 15, 2025	Revision 0 Design Document	Updates to the design document to include all design changes during development	Updated Design Document
February 3, 2025	Revision 0 Demonstration	Demonstration of current progress in development	Received feedback from stakeholders
March 7, 2025	Revision 0 V&V Report	Updated testing plan to reflect one actually used during development	Updated V&V Report
March 30, 2025	Revision 1 Final Demonstration	Demonstration of our final working product that reflects all the requirements of the stakeholders	Final Product is working as intended
April 8, 2025	EXPO Demonstration	Demonstration of our final product in the form of a presentation and answer any question asked by the audience	Successfully present and display our final product
April 2, 2025	Revision 1 Final Documentation	Final Revision of all documentation that is up to date and includes all design changes made after revision 0	Complete final documentation

(P.5) Required technology elements

- **Point Cloud Library** : The point cloud library provides a library of algorithms for 3D geometry and point cloud processing. This is essential for working with point cloud data.
- **Open CV Library** : The Open CV library provides a wide variety of algorithms to assist in human detection and is an industry standard when it comes to computer vision.
- **Microsoft Kinect** : The microsoft Kinect is necessary for producing usable data that can be analyzed.
- **Kinect2Grabber** : Kinect2Grabber is a provided tool that retrieves point cloud data from a Kinect v2 into PCL.

(P.6) Risk and mitigation analysis

Some potential risks during development that can prevent certain dates in the schedule from being met are listed below:

- **Unforeseen circumstance preventing team member from completing assigned task** : In the event that an unforeseen circumstance causes a team member to not be able to complete their work on time, they should immediately inform the team and agree to take on extra responsibilities for future tasks to make up for it. The rest of the team should accommodate the team member and do their best to help meet deadlines. This risk can be minimized by setting weekly goals and having a meeting weekly to discuss progress on work. This enforces team members to work on their assigned work continuously over the whole period they are given to work on it.
- **Unforeseen technical issues that prevents a working product to be demonstrated or submitted** : Not being able to create a reliable algorithm, software bugs, hardware failure or compatibility issues can all delay progress in the project and could happen before a code demonstration or submission. We must ensure to always have working builds we can rollback to in case of certain features not being completed in time before needing to demonstrate or submit our product. Additionally, we must ensure to set realistic goals and allocate enough time during our development plan for testing, debugging and integration. Testing should be done in every stage of development so that there would always be a version we of the code we can rollback on and issues are found early on.

(P.7) Requirements process and report

1. The initial requirements elicitation process begins with reading and translating the initial project description written by Dr. Gary Bone.
2. As our only stakeholder, Dr. Gary Bone will be interviewed through a meeting with the entire development team. The objective is to find out the motivation behind the project, define specific requirements, get clarifications and ask general questions.
3. We will communicate regularly with Dr. Bone throughout the development process and the requirements will be refined and updated. Any new requirements brought to our attention by Dr. Bone will be added to the requirements and we will communicate with Dr. Bone if we have any

questions and update the requirements document accordingly.

Likely Changes

1. **Offline Processing** [F413]: The offline aspect is not an important fucntional requirement. The main focus with the software is to be able to pick up live data.
2. **Body Pose Variation Handling** [F414]: There may be some poses that the system may not be able to handle and so there may need to be adjustments to this requirement.
3. **Usability** [NF431]: Usability is not the main focus when it comes to the software and thus could be removed.

Unikely Changes

1. **Human Detection** [F411]: The main requirement for this system is human detection and so this requirement will not change.
2. **Location Estimation** [F412]: The location prediction was a newly added requirement and needs to be implemented to showcase the hidden human detection feature in the software.
3. **Integration with Kinect Sensor** [F415]: The Kinect Sensor is a constraint and so the software will need to be implemented through the data collected by the Kinect stream.
4. **Real-Time Processing** [F416]: The data needs to be processed within real time to provide that the output is useful.
5. **Reliability** [NF432]: The final product will always need to be reliable to ensure that the humans location is correctly identified.
6. **Accuracy:** [NF433]: The accuracy of the software needs to be held to a high standard to ensure that every time the user runs they expect a similar result.

Reflection

1. *What went well while writing this deliverable?*

   The work was divided up and everyone was able to complete on time. Communication between the team members was efficecnt and worked out well to help each other on the questions. Some sections overlapped so communication was very important for this assignment.

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

   The biggest pain points may have been trying to make appropriate components and also making sure that the functional and nonfunctional requirements made sense for each case.

3. *How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?*

   Majority, if not all of the functional requirements were through our meetings with Dr.Bone and our meetings. The non functional requirements were a mix of expectations for the project and past projects, plus the requirements course.

4. *Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project?*

   The course that helped with building the goals and requirements has been the software requirements course. An important step when it comes to working on big projects is really understanding the project to the core before anything is done and thats what this course helped with. Another course would probably be our testing course which will really help when testing the software (unit tests and functionality tests). Finally, the multiple coding classes that were taking for example OOP that provided knowledege on how to write clean code and how to implement multiple alogorithms.

5. *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.

6. *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