Cluster Tracker

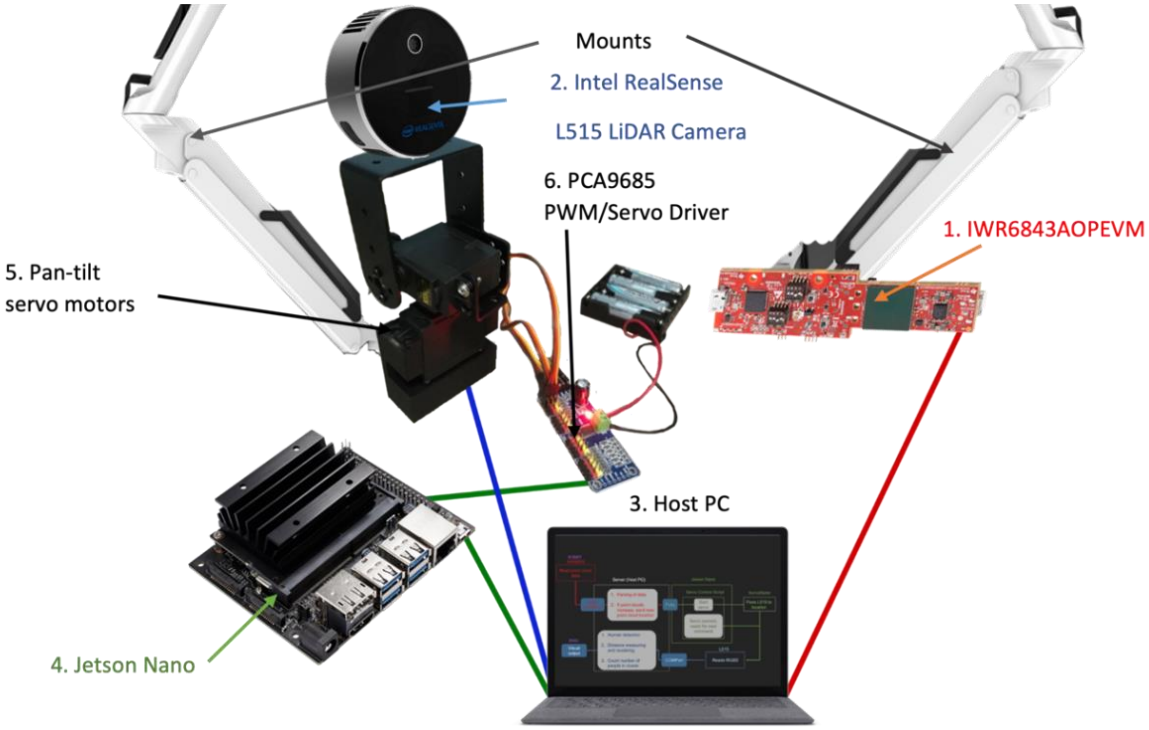
The *Cluster Tracker* is a sensor system designed to track clusters of people through sensor fusion and artificial intelligence. It aims to alleviate the manpower required to enforce social-distancing rules, however, minor adjustments can be made for the system to perform other tasks.

Targets' metadata obtainable by the sensors onboard the system are:

1. Velocity of targets relative to the system
2. Acceleration of targets relative to the system
3. Range (x, y, z distances) of targets relative to the system
4. Azimuth and elevation of targets relative to the system
5. Number of targets present in the Field-Of-View (FOV) of the system

Key Components:

1. IWR6843 antenna-on-package (AOP) evaluation module (EVM) from Texas Instrument
2. Intel RealSense L515 LiDAR Camera
3. Host PC
4. NVIDIA Jetson Nano Developer Kit
5. MG995 Servo Motors
6. PCA9685 Servo Driver



General specifications:

FOV (horizontal)	120°
Range	12 m
Maximum velocity	6 m/s
Max number of targets	13
Accuracy*	62%

*Refers to the average number of violations detected by the Cluster Tracker over the total number of actual violations.

Goh Kheng Xi, Jevan (A0199806L)
Ryan Tang Li Yen (A0200900A)
Lock Mei Lin (A0204751M)

Cluster Tracker

Operational Flow

The radar (IWR6843 AOP EVM) captures and sends point clouds of the environment to the server script (on the Host PC) which clusters the points into objects, performing preliminary checks for moving people spaced at less than one metre apart. The location of a possible violation of this social-distancing rule is then sent to the Jetson Nano, which pans the LiDAR camera (L515) so that it tracks this cluster of people with its higher resolution. The latter's RGB-Depth data is propagated through an object detection and classification model on the Host PC, after which exact locations of each target are produced and used to more accurately check for any violations of social-distancing rules.

NVIDIA Jetson Nano

In its fully-fledged form, the Cluster Tracker is envisioned to be deployed without the direct intervention of the Host PC, such that all data processing and servo motor control are done on the Jetson Nano. In our prototype, however, we have yet resolved the ARM architecture incompatibility with the pre-compiled sensor software; therefore, the Nano serves mainly as a microcontroller. Using socket programming, we interface with the Nano wirelessly from the Host PC.

Texas Instruments IWR6843 mmWave Sensor

The IWR6843 AOP EVM is a millimetre wave radar that operates at around 60GHz. Due to the relatively higher frequency and thus, shorter wavelength compared to other radars, it has a higher accuracy but lower range. However, as radars in general have too low of a resolution to discern single entities in close proximity, we complement it with a secondary sensor (L515).

Using complex signal processing techniques, clustering algorithms, and state estimations such as DBScan and the extended Kalman Filter, point clouds are parsed into targets whose metadata are then received by the server which will in turn inform the L515 where is the target located through the Jetson Nano.

Intel RealSense L515 LiDAR Camera

The L515 is a time-of-flight depth sensing 3D camera that uses an infrared light to collect depth data. The time taken for the light to return to the camera is used to calculate the distance of the object away from the camera. By repeating this process millions of times a second, the L515 can build a high-resolution depth image. Each pixel in the depth image represents the distance between the image plane and the corresponding object in the depth image. When the depth frame is aligned to the colour frame, the depth value can be obtained from the RGB pixel instead.

For our purposes, the YOLOv5 object detection model is used for human detection from the RGB image, returning the 2D pixel coordinates of the human which will be used to draw a bounding box around each detected human on the aligned frames. The distance between each detected human can then calculated using deprojection which takes a 2D pixel coordinates on a stream's images, as well as corresponding depth and intrinsic camera parameters to map it to a 3D point location within the stream's associated 3D coordinate space.