Person near wall makes clustering fail

This test case is based on observation of an issue where the person would no longer be detected, often when the person was close to a wall or other large object.

Scenario:

- Stand in front of the robot far from other objects, make sure person is detected;
- Person slowly walks closer and closer to the wall until they are no longer detected.

We performed this scenario in one of the small work rooms, with only one dynamic person present in the lidar's field of view. The scenario was performed 5 times with various angle of approach to the wall, and to a door.

Observations:

It seems that the person tends to no longer be detected once they get closer than about 0.1-0.05 meters. When checking the various markers on Rviz, we saw that the motion detection was fine, but the person's legs were not marked as being legs. The step between these is the clustering. When displaying the cluster start and end points, we noticed that when the person got too close to the wall, instead of their leg being it's own cluster, it was merged into a single larger cluster including the nearby object (e.g. wall, cupboard, door), as shown in Fig.1 .

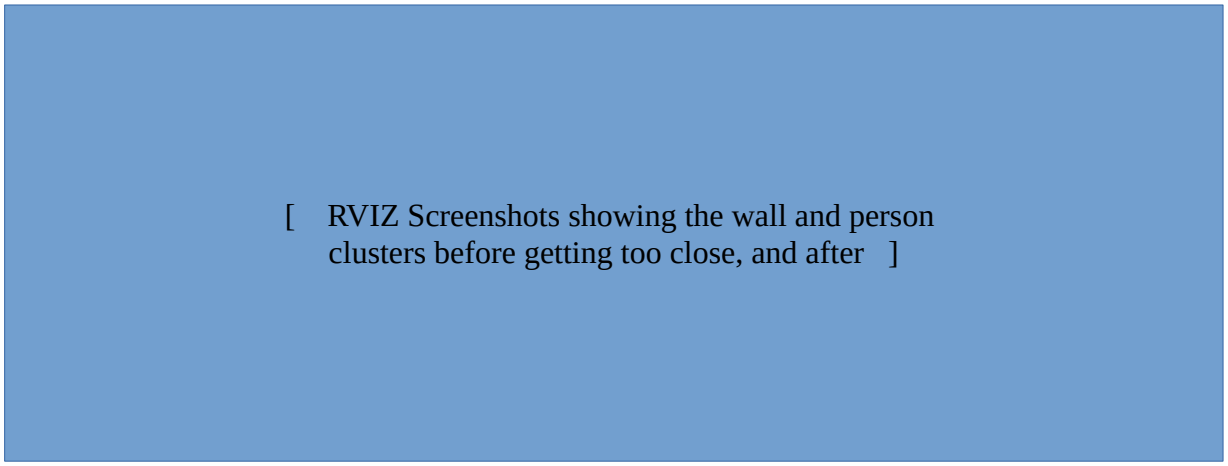

Fig. 1 Rviz screenshots of person approaching a wall. Left: leg and wall clusters are distinct. Right: leg and wall are merged into a single cluster after approaching the wall.

Analysis:

If the size of this merged cluster is larger than the leg_max_size we use in the leg detection step, then the person can no longer be detected. The two clusters being merged together is due to the clustering metric we use, which only considers a new cluster should be formed if the next consecutive point is more than 0.05m away from the previous point. In our code, the condition for creating clusters based on this threshold is at line 86 in datmo.cpp.

Improvements:
Making the cluster threshold smaller would make the leg and wall merge less often, requiring them to be closer. However it would also lead to objects being broken into several clusters when they shouldn’t, such as wall sections which are far away, which tend to be represented by few laser hits with wider spacing.
We could try to use more information to make a more complex clustering condition, such as a clustering threshold which changes based on the range of the laser hits (higher for further hits).

Opposite direction (working from the algorithm instead of from an observation)
This test case is based on an analysis of the clustering method, specifically the condition used to determine when one cluster ends and the next one starts.

Algorithm analysis:
We have a parameter called clustering threshold. It is used to determine when we will consider that two points are part of different clusters. In our code, the condition for creating clusters based on this threshold is at line 86 in datmo.cpp. We simply check if the euclidean distance between consecutive laser hits is greater than a fixed threshold, in our case, 0.05m. If two things in the real world move closer until they are closer than this threshold, they will be considered part of a single cluster. If one of those is a person's leg, and the other cluster is large enough, then the final merged cluster will be bigger than our leg max size. In that case, the person's leg is not considered as a leg because it's cluster is too large, so the person cannot be detected.

Scenario:
To trigger this execution path or behaviour, we can follow these steps:
- make sure person is detected, far from other objects;
- person moves closer and closer to a wall or other large object until they are no longer detected.

We performed this scenario in one of the small work rooms, with only one dynamic person present in the lidar's field of view. The scenario was performed 5 times with various angle of approach to the wall, and to a door.

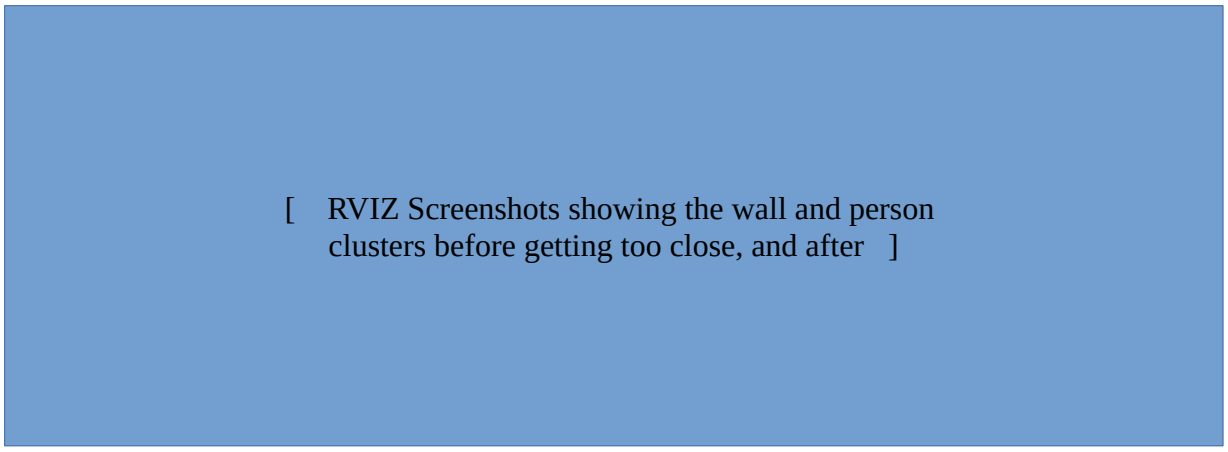

Fig. 1 Rviz screenshots of person approaching a wall. Left: leg and wall clusters are distinct. Right: leg and wall are merged into a single cluster after approaching the wall.

Observations:
We see that before the person gets too close, their leg and the object are still marked as two different clusters, and the leg marker is properly displayed. After the person gets too close, using the cluster start and end markers we see that their leg and the wall are considered as one large cluster, and the leg marker disappears, leading to the absence of person detection.

Improvements:
Making the cluster threshold smaller would make the leg and wall merge less often, requiring them to be closer. However it would also lead to objects being broken into several clusters when they shouldn’t, such as wall sections which are far away, which tend to be represented by few laser hits with wider spacing.
We could try to use more information to make a more complex clustering condition, such as a clustering threshold which changes based on the range of the laser hits (higher for further hits).