Long description of the competing system

UWB (Ultra-Wide Band) has gained wide attention and has been increasingly recognized as a promising localization technology. Owning its high bandwidth, it can achieve high-precision ranging and positioning. However, its performance significantly degrades in practice due to possible low Signal-To-Noise ratios (SNRs) and Non-Line-Of-Sight (NLOS) issues. The real-time requirement of competition also is a challenge.

The TOF-based estimation algorithm determines the positioning accuracy. The most common TOF-based estimation algorithms can be broadly divided into threshold comparison, peak detection, iterative peak subtraction (IPS), and leading-edge detection.

Since the data is recorded using a platform based on the Decawave DW1000 UWB chip, we have referred to the corresponding manuals and proposed a method to estimate TOF and obtain distance estimation based on leading edge detection algorithms, and Fig. 1 shows the pipeline of our algorithm:

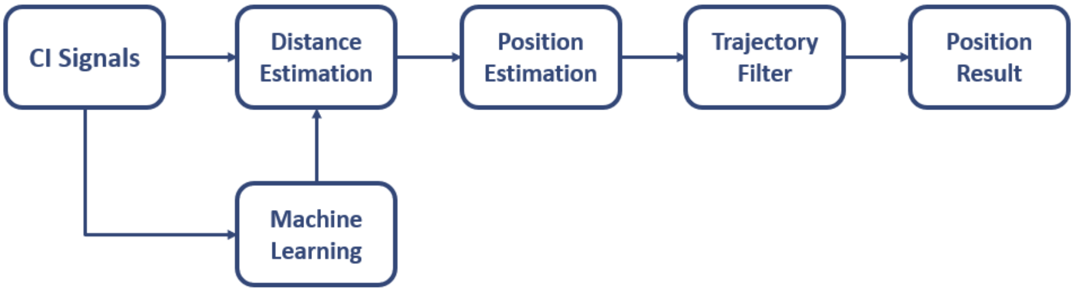

Figure 1 Algorithm Pipeline

At first glance, the result of the algorithm looks promising, but the distance estimation is very coarse, and there still exist some obvious estimation errors. Distance obtained by rule-based algorithm also cannot meet our high accuracy requirements due to the distance resolution of CI data provided by the competition is relatively low. What's more, some ci-signals can hardly extract the accurate distance between transmitter and receiver in low SNRs or NLOS conditions.

To achieve a more accurate positioning result, we try to introduce a neural network method to solve these problems mentioned above. We have designed many hand-crafted features, e.g., the output of modified leading edge detection algorithms and distance corresponding to the highest peak in the CI, as the input to estimate the distance between transmitter and receiver. Meanwhile, we use evaluation criteria such as $IDiff$ to measure the probability of whether the estimation was made in an NLOS or low SNRs situation.

$$IDiff = |\ \text{first path position - peak path position}\ | \qquad (1)$$

$$peakAmp \ = \max(Amp_{CI}) \qquad (2)$$

Next, we collect the distances from anchors with the same burst-id to estimate the target's position. We take the anchor as the center of a circle and the estimated distance as the radius. Candidate positions can be obtained for eight anchors based on the trilateration algorithm. For all these alternative positions, we apply the clustering algorithm to obtain the cluster center, which is the estimated position of the target. Figure 2 shows an example of our method to get an estimated location.

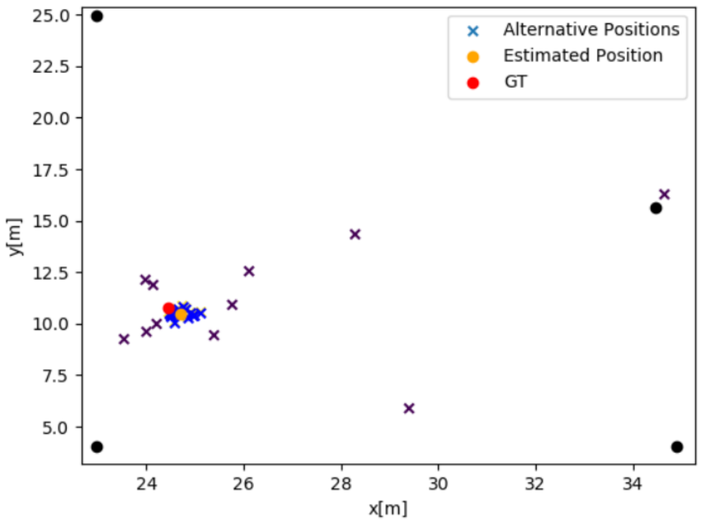

Figure 2 Get estimated **position by cluster algorithm**

Finally, we use the SG filter to smooth the trajectory, resample it to get the final positions, and take the Euclidean distance between estimated and accurate results as the primary evaluation metric.

Now we have been exploring and improving our method. We aim to find an end-to-end approach to solve this problem efficiently and accurately.