Enhancing Doppler Ego-Motion Estimation: A Temporally Weighted Approach to RANSAC

Samuel Lovett, Kade MacWilliams, Sreeraman Rajan, Carlos Rossa
Systems and Computer Engineeing
Carleton University, Ottawa, Canada
{samuellovett; kademacwilliams}@cmail.carleton.ca, {sreeramanr; rossa}@sce.carleton.ca

***Abstract*—Ego-motion estimation from millimetre-wave Doppler are subjected to high levels of outliers caused by multipath reflections. A common method to reduce the effect of outliers is to fit measurement points to a velocity model using the random sample consensus (RANSAC) algorithm, where the velocity model is traditionally parameterized from data collected from a single time step. In this paper, we show that the temporal relationship of data points between successive measurements can be exploited to improve velocity estimation. Two variations of RANSAC with an integrated weighted sliding window are proposed. Each point in the window receives a weight that decreases over time. In the first algorithm, points are sampled with a temporally weighted probability and the velocity model is generated using least square regression (LSQ). In the second algorithm, points are sampled with a uniform probability but the model is parameterized with a temporally weighted LSQ. The motion of the platform is then calculated for both methods. Experimental results using data from three indoor locations demonstrate an average position accuracy improvement of 19.5% over conventional RANSAC implementations.**

***Index Terms*—Ego-motion, mmWave, weighted RANSAC, weighted least-squares, sliding window.**

I. Introduction

How does a sensing platform know how far it has travelled? This question has been one of the forefront topics in autonomous vehicle research. A popular approach to answer this question is through ego-motion estimation, that is, the process of calculating the platform's motion relative to a scene. In ego-motion, the motion a sensor observes is either caused by the sensor's self (*ego*) motion or by objects moving in the sensor's scene. When the scene is static, this relative motion is used to infer the platform's position over time [1].

The most common sensors used in mobile platforms include LIDARs, cameras, and radars. Radar sensors, such as frequency modulated continuous wave millimetre wave (mmWave sensors) have a number of advantages over LIDAR and cameras. A longer wave length allows them to operate in extreme conditions where LIDAR and cameras typically fail, such as environments with low viability due to smoke, dust, or fog [2]–[4]. Additionally, FMCW mmWave sensors can measure the velocity of objects via the Doppler effect [5].

As mmWave sensors transmit electromagnetic waves, they are reflected off objects and returned to the sensor. However, these waves do not always take a single path back. Waves often reflect off multiple objects, and when returned to the sensor lead to "ghost points" within the measurement. So called because they do not represent a real object. As such, these points due to multipath reflections must be treated as outliers and discarded. In [6] it was found that for a single indoor measurements, up to 75% of all points observed can be outliers. In addition, the data collected by mmWave sensors is very sparse compared to data collected by LIDARs [6]. Combined, data sparsity and multi-path reflections result in measurements with a high ratio of outliers to inliers.

These challenges entail that mmWave sensors cannot replace LIDAR sensors in existing ego-motion estimation methods. For example, since LIDAR measurements are denser [6], ego-motion can be estimated by comparing the relative rotation and translation between consecutive measurements in a process known as scan-matching. With mmWave measurements, additional deep-learning processing is often used [4], [6], [7]. However, these methods require extensive training and their ability to generalize to different locations is limited.

Another ego-motion estimation approach with mmWave is to use the Doppler velocity of the sensor's surroundings to estimate its motion [8]–[10]. This is done by repeating 3 iterative steps for each measurement: 1) fit a velocity model to the measurement, 2) use the model to remove multipath reflections from them, and 3) rerun step 1) without outliers to refine the model. Once the model is parametrized, the velocity is integrated to calculate the platform's motion. As noted in [9], the accuracy of these methods depends heavily on the model's ability to identify and reject outliers (points due to multi-path reflections). With a large number of outliers, the velocity estimate may not converge to the true platform velocity, causing the position error to drift over time.

To remove measurement outliers during velocity model fitting, different algorithms have been proposed. A commonly used algorithm is the random sample consensus (RANSAC), an iterative model fitting algorithm designed for data containing outliers [8], [11]–[15]. For each measurement, the RANSAC algorithm randomly samples several Doppler velocities to fit the model. Then the error between the model's estimated Doppler and the measured Doppler is calculated, and based on this error, points are categorized as inliers or outliers. The process of selecting points, fitting the model, and calculating the error is then repeated a set number of times.

This research is supported by the Queen Elizabeth II Scholarship in Science and Technology. Cette recherche est appuyée par la bourse d'études supérieures en sciences et technologie de la reine Élizabeth II.

The model with the largest number of inliers and lowest error is selected. When the measurement has more outliers than inliers, model parametrization is not accurate [8].

Scan matching for LIDAR is based on the principle that consecutive measurements are correlated over a short period of time by the platform's motion [16]. Applying this logic to mmWave measurements, the Doppler velocity of stationary objects (inliers) should be consistent over short periods of time. Thus, we can expand on previously proposed methods by hypothesizing that this temporal relation between inliers does not hold for outliers, i.e., outliers are errors unique to the time of their measurement, and independent in time with the motion of the platform [17].

Provided that this hypothesis is true, a sliding window concatenating multiple measurements should increase the density of only the inlier points, while leaving the outlier density approximately the same. In this paper, to verify this hypothesis, we propose two new variations of RANSAC, which estimate ego-motion over an integrated weighted sliding window of m previous measurements. The sliding window of measurements are concatenated into a first-in-first-out temporal queue, as shown on the left in Fig 1. The location of the measurement within the queue defines the weight that each point within the measurement receives. In the first method, Temporal Sample Consensus for ego motion estimation (TEMPSAC), the velocity model is fit to measurement points with higher weights, biasing the velocity model in time, as seen in the top of Fig. 1. In the second method, Temporally Weighted Least-Squares for ego motion estimation (TWLSQ), the weights are used for a weighted least-squares regression to improve velocity model fitting, as seen in the bottom of Fig. 1. In this way, both methods capture the temporal relationship of ego-motion, resulting in more accurate motion estimation compared to past RANSAC implementations. To the best of our knowledge, this is the first application of a temporally weighted moving window RANSAC for ego-motion estimation with radar data.

The paper is structured as follows: In Section II an overview of the proposed algorithms is presented. The algorithms are then optimized using the cost function defined in Section II-D, after which they are experimentally validated in Section III using the open source radar dataset Coloradar [18]. In total, 300 full path evaluations are conducted to compare our methods against a standard implementation of RANSAC, initially proposed in [8]. The results show that both methods increase the accuracy of ego-motion estimation by over 19%. The algorithms proposed in this paper are publicly available[1].

II. Improved Ego-Motion Estimation with Weighted Sliding Window RANSAC

To introduce the proposed algorithm, Section II-A first details how motion is estimated assuming the mmWave Doppler velocity data is outlier free. Section II-B describes how the standard RANSAC algorithm is used to remove outliers from non-ideal data. Finally, Section II-C describes how the proposed methods leverage the temporal relationship of mmWave measurements to improve motion estimation.

Hereafter a matrix is written in bold uppercase, a vector in bold lowercase, a scalar in unbolded lowercase, and their superscript defines the reference frame in which they are being expressed. Further, we consider a mobile platform equipped with mmWave radar and define three reference frames as shown in Fig. 2A. The fixed world reference frame RF_w is where the ego-motion of the platform's body frame RF_b is observed. Attached to the RF_b is the sensor reference frame RF_s where all mmWave measurements are generated.

A. Ego-Motion Estimation using mmWave Radar

A mmWave sensor moving in a static environment can be equivalently represented from the sensor's perspective as a stationary mmWave sensor with moving environment. Using this representation, the environment will have a relative motion equal but opposite to the sensor's original motion. We will use this frame of reference moving forward. Now consider that at sample i, the mmWave measures a cluster of points ${}^s\mathbf{M}_i$, i.e., a collection of a_i data points $\mathbf{p}_{j=1,\dots,a_i}$ such that

$$ {}^s\mathbf{M}_i = \begin{bmatrix} {}^s\mathbf{p}_j & \dots & {}^s\mathbf{p}_{a_i} \end{bmatrix} \tag{1} $$

$$ {}^s\mathbf{p}_j = \begin{bmatrix} {}^s\theta_j & {}^s v_{d_j} \end{bmatrix}^T \tag{2} $$

where ${}^s\theta_j$ is the angular location, and ${}^s v_{d_j}$ is the Doppler speed of the j^{th} point ${}^s\mathbf{p}_j$ in RF_s, respectively, as illustrated in Fig. 2A. The Doppler speed ${}^s v_{d_j}$ is the component of the object's velocity (-${}^s v_s$) in the radial direction. In other words, the component along the line projecting from the object to the centre of RF_s at the angle ${}^s\theta_j$ with respect to the horizontal axis ${}^s\mathbf{x}$ of RF_s. As the platform moves through the environment with a constant speed, the measured Doppler speed of static objects will change depending on their location. This is shown in Fig. 2A when the platform moves from sample $i = 1$ to $i = 2$. When multiple objects are observed within one measurement, as shown by the black circles in Fig. 2B, Kellner *et al.* [8] showed that the Doppler speed has a sinusoidal relationship with the sensor's speed ${}^s v_s$. By transforming these measurements from the ${}^s\mathbf{x}$-${}^s\mathbf{y}$ plane to the ${}^s\theta$-${}^s v_d$ domain, as in Fig. 2C, the object's Doppler speed can be modelled as,

$$ {}^s\hat{v}_{d_j} = -{}^s v_{s_i} \cos\left({}^s\theta_j - {}^s\alpha_i\right) \tag{3} $$

where ${}^s\hat{v}_{d_j}$ is the estimated Doppler speed of the model, which is parameterized using ${}^s\alpha_i$ and ${}^s v_{s_i}$. Here, ${}^s\alpha_i$ is the angular direction of the sensor's speed ${}^s v_{s_i}$ at sample i. Expanding (3) the model becomes,

$$ {}^s\hat{v}_{d_j} = -{}^s v_{s_i} \left[\cos({}^s\theta_j)\cos({}^s\alpha_i) + \sin({}^s\theta_j)\sin({}^s\alpha_i)\right]. \tag{4} $$

To solve for the unknown ${}^s\alpha_i$ and ${}^s v_{s_i}$ parameters in (3), Eq. (4) can be reformulated as a matrix equation:

$$ \begin{bmatrix} {}^s v_{d_1} \\ \vdots \\ {}^s v_{d_j} \end{bmatrix} = \begin{bmatrix} \cos {}^s\theta_1 & \sin {}^s\theta_1 \\ \vdots & \vdots \\ \cos {}^s\theta_j & \sin {}^s\theta_j \end{bmatrix} \begin{bmatrix} {}^s v_{s_i} \cos({}^s\alpha_i) \\ {}^s v_{s_i} \sin({}^s\alpha_i) \end{bmatrix}. \tag{5} $$

[1] https://github.com/samuelLovett/tempEgo.git

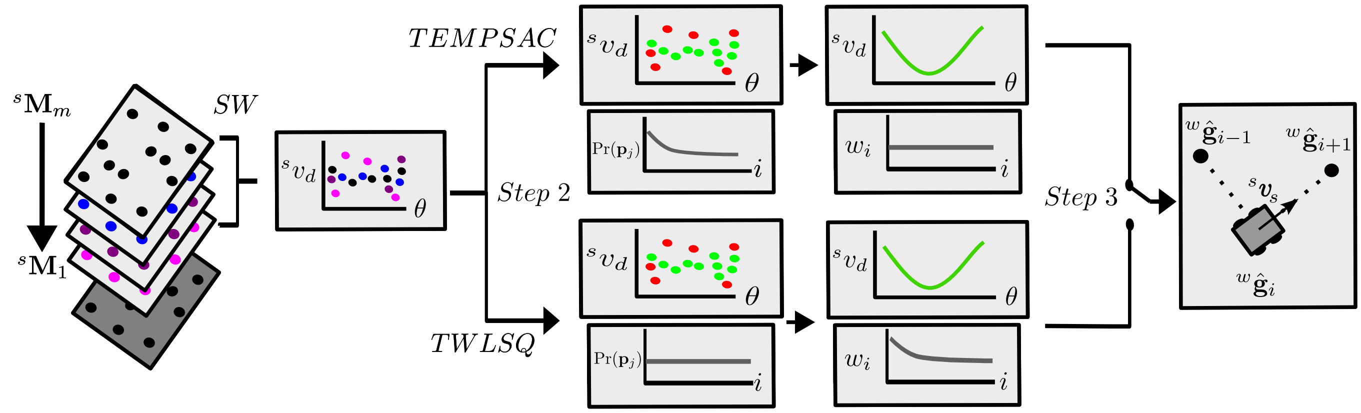

Fig. 1. The proposed ego-motion estimation methods. Measurements are placed within the sliding window and then combined. In TEMPSAC, points are sampled with a temporally weighted probability and then the velocity model is generated using LSQ. In TWLSQ, the points are selected having a uniform probability but the model is parameterized with a temporally weighted LSQ. The motion of the platform is then calculated for both methods.

containing all objects $j = 1, \ldots, a_i$ in ${}^s\mathbf{M}_i$, and solved using the measured Doppler velocities via least-squares regression (LSQ). The resulting ${}^s\alpha_i$ and the sensor speed ${}^s v_{s_i}$ are then converted to the Cartesian vector, ${}^s\mathbf{v}_{s_i}$. Next, to calculate the motion of the platform, ${}^s\mathbf{v}_{s_i}$ must be transformed into RF_w through a series of homogeneous transformations,

$$^w\tilde{\mathbf{v}}_{s_i} = {}^w\mathbf{H}_{b_i}{}^b\mathbf{H}_s{}^s\tilde{\mathbf{v}}_s \tag{6}$$

where ${}^s\tilde{\mathbf{v}}_s$ is the homogeneous form of ${}^s\mathbf{v}_s$ such that ${}^s\tilde{\mathbf{v}}_s = [{}^s\mathbf{v}_s \quad 1]^T$, ${}^b\mathbf{H}_s$ is the homogeneous transformation from RF_b to RF_s governed by rotation angle ${}^b\beta$ between ${}^s\mathbf{x}$ and ${}^b\mathbf{x}$ about ${}^s\mathbf{x} \times {}^s\mathbf{y}$, and ${}^w\mathbf{H}_{b_i}$ is the homogeneous transformation from RF_w to RF_b governed by the rotation angle ${}^w\phi_i$ between ${}^b\mathbf{x}$ and ${}^w\mathbf{x}$ about ${}^w\mathbf{x} \times {}^w\mathbf{y}$ at i (see Fig. 2A). With ${}^w\mathbf{v}_s$ known, the platform's current position is the discrete integration of velocities from the first sample $i = 1$ to the current sample i,

$$^w\mathbf{d}_i = \sum_{i=1}^{i} {}^w\mathbf{v}_{s_i} \cdot \Delta t_i \tag{7}$$

where Δt_i is the change in time between the current and past measurement sample (i.e., i and $i-1$).

The estimated pose of the platform ${}^w\hat{\mathbf{g}}_i$ is defined as

$$^w\hat{\mathbf{g}}_i = [{}^w\mathbf{d}_i \quad {}^w\phi_i]^T. \tag{8}$$

B. Random Sample Consensus (RANSAC) Filtering

The previous section assumed that all points in the measurement are inliers (static and not ghost points). To fit the velocity model using measurements containing a large amount of outliers, filtering is required. RANSAC is a model estimation algorithm robust to data with many outliers [11], [19]. The algorithm has three steps, 1) estimate model parameters, 2) partition data, and 3) determine best partition and refit model.

Step 1 - Estimate model parameters: Given a measurement ${}^s\mathbf{M}_i$ of a_i points ${}^s\mathbf{p}_j$, a *random* selection of n points from ${}^s\mathbf{M}_i$ can be used to find ${}^s\alpha_i$ and ${}^s v_{s_i}$ by solving (5) through LSQ regression [11]. In the case of a cosine curve, the minimum number of points to fit the curve is $n = 2$. The random selection of an entry ${}^s\mathbf{p}_j$ is governed by the uniform probability distribution such that,

$$\mathrm{Pr}({}^s\mathbf{p}_j) = \mathcal{U}_{[1,a_i]}, \tag{9}$$

where $\mathrm{Pr}(\cdot)$ is the probability function of a random variable.

Step 2 - Partition the data: For a point ${}^s\mathbf{p}_{j=1,\ldots,a_i} \in^s \mathbf{M}_i$, the error e_j between the measured point's Doppler speed ${}^s v_{d_j}$ and the speed estimated from the model fit in Step 1 ${}^s\hat{v_{d_j}}$ is,

$$e_j = ({}^s v_{d_j} - {}^s\hat{v_{d_j}})^2. \tag{10}$$

A point ${}^s\mathbf{p}_j$ is considered an inlier if the error is below a predefined threshold ε, that is:

$$inliers = \{{}^s\mathbf{p_j} \mid e_j < \varepsilon\}. \tag{11}$$

The number of inliers c is then counted. If c is larger than a predefined threshold z, then the algorithm proceeds to Step 3, otherwise it goes back to Step 1.

Step 3 - Determine best partition and refit model: LSQ is then repeated to update the parameters in (3) using all identified inliers. The error in (10) is re-evaluated for all inliers. To determine the quality of the refined model, the mean error of all inliers is calculated:

$$\bar{e} = \frac{1}{c}\sum_{j=1}^{c} e_j \qquad {}^s\mathbf{p_j} \in inliers. \tag{12}$$

This 3-step process is repeated k times. The model parameters that result in the lowest error are used to parametrize (3), which is then used to determine the platform's motion ${}^w\mathbf{d}_i$.

C. Ego-Motion Estimation Using Temporal Weighting

In order to leverage the temporal dependency of inliers and the temporal independence of outliers between measurements, a first-in-first-out sliding window **SW** with m measurements is defined. Henceforth, i is the index within SW such that $\mathbf{SW} = \{{}^s\mathbf{M}_{i=m}, \ {}^s\mathbf{M}_{i=m-1}, \ldots, \ {}^s\mathbf{M}_{i=1}\}$, as shown on the left of Fig. 1, where the newest measurement is ${}^s\mathbf{M}_m$ and the oldest is ${}^s\mathbf{M}_1$. Equation (2) is redefined as

$$^s\mathbf{p}_j = \left[{}^s\theta_j \quad {}^s v_{d_j} \quad w_i\right]^T \qquad \forall \quad {}^s\mathbf{M}_i \in \mathbf{SW}. \tag{13}$$

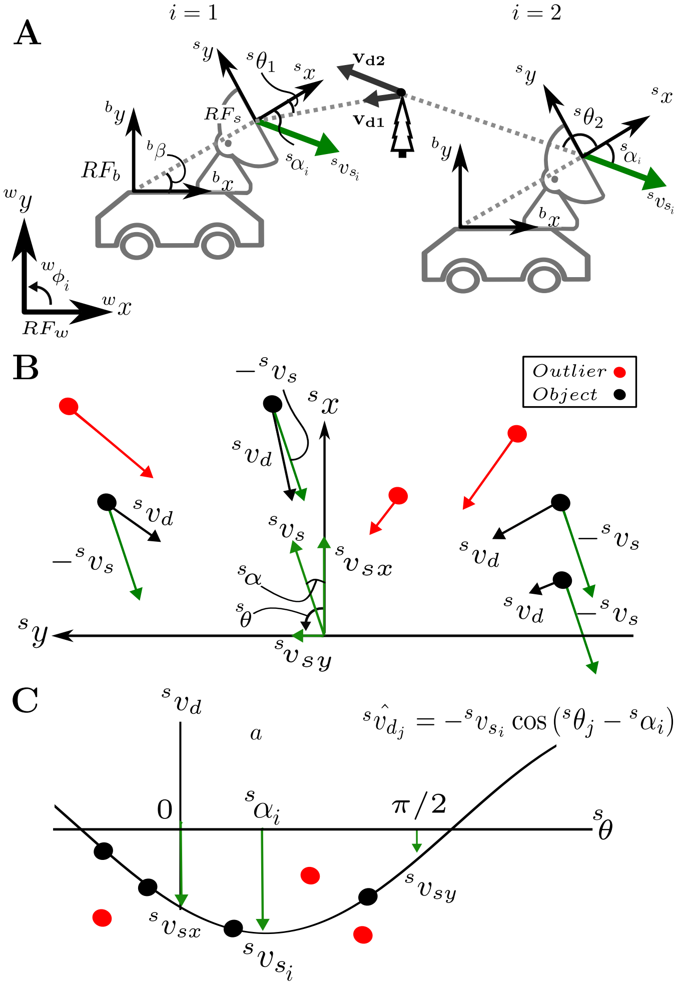

Fig. 2. **A)** Doppler velocity measurement ${}^s\mathbf{v}_{d_i}$ from a mobile platform moving with speed ${}^s v_s$ at an angle ${}^s\alpha$ with respect to (w.r.t) ${}^s\mathbf{x}$. At $i=1$ and $i=2$, the sensor measures ${}^s\mathbf{v}_{d_1}$ and ${}^s\mathbf{v}_{d_2}$, respectively, in the sensor reference frame RF_s along ${}^s\theta$ w.r.t. ${}^s\mathbf{x}$. The orientation of the body reference frame RF_b w.r.t to the world reference frame RF_w, and of RF_s w.r.t RF_b, are given by ${}^w\phi$ and ${}^b\beta$, respectively. **B)** Example of a mmWave measurement ${}^s\mathbf{M}_i$ with all static points shown in black and outlier points shown in red. In RF_s objects appear moving with velocity $-{}^s v_s$. Here the i and j subscripts have been omitted for readability. In **C)** points from **B)** are viewed in the ${}^s\theta$-${}^s v_d$ domain. The object points fit a sinusoidal curve ${}^s\hat{v}_{d_j} = -{}^s v_{s_i}\cos({}^s\theta_j - {}^s\alpha_i)$ from which the platform's velocity is estimated.

where w_i is the sliding window index-dependent weight assigned to all points ${}^s\mathbf{p}_{j=1,\dots,a_i}$ in ${}^s\mathbf{M}_i$ and calculated as,

$$w_i = \frac{\lambda^{m-i}}{\sum_{i=1}^{m}\lambda^{m-i}} \qquad \lambda \in [0,1] \tag{14}$$

where $0 \leq \lambda \leq 1$ is the user defined fixed forgetting factor as in [20].

As $\lambda \to 0$, more weight is placed on newer measurements points than old points. As $\lambda \to 1$, the points are weighted equally. Unlike in [20], $w_i \in [0,1]$ so it can be thought of as the probability of forgetting the points in ${}^s\mathbf{M}_i$, $\Pr({}^s\mathbf{M}_i) = w_i$.

The weights dual usage in the sliding window is the basis of our proposed algorithms: When the weights are used as probabilities, they define the likelihood of a point in **SW** being selected to fit the model. When used in a weighted LSQ regression, the temporal location of ${}^s\mathbf{M}_i$ is considered. These two RANSAC algorithms will hereafter be referred to as the Temporal Sampling Consensus (TEMPSAC), and the Temporally Weighted Least-Squares (TWLSQ).

Temporal Sampling Consensus for Ego Motion Estimation (TEMPSAC): The temporally weighted points in (13) are used for Step 1 in the regular RANSAC algorithm (see Section II-B), where the probability of selecting a point is governed by (9). In contrast, here this probability is replaced with

$$\Pr({}^s\mathbf{p}_j) = \frac{w_i}{a_i} \qquad {}^s\mathbf{p}_j \in {}^s\mathbf{M}_i. \tag{15}$$

Thus, more recent points are more likely to be used. To summarize the algorithm for each sample i: 1) The sliding window and weights are updated (see Fig. 1, left), 2) TEMPSAC is calculated over the sliding window, 3) model parameters with the lowest error are used in (3) to calculate ego-motion.

Temporally Weighted Least-Squares for Ego Motion Estimation (TWLSQ): The temporally weighted points in **SW** (13) are used for Step 1 in the regular RANSAC algorithm presented in Section II-B. In contrast to regular RANSAC, a weighted-LSQ (WLSQ) regression is used for model fitting (Step 1) and evaluation (Step 3), and (10) is replaced with

$$e_j({}^s\theta_j) = w_i({}^s v_{d_j} - {}^s\hat{v}_{d_j})^2 \qquad \forall\ {}^s\mathbf{M}_i \in \mathbf{SW}. \tag{16}$$

By applying the WLSQ regression a higher weight is given to the errors observed in more recent measurements, while still sampling uniformly across the sliding window. In summary, for each sample i: 1) The sliding window and weights are updated (see Fig. 1 left), 2) a model is fit to the window and WLSQ regression is calculated, 3) model parameters with the lowest error are used in (3) to calculate ego-motion.

D. Hyperparameter Optimization

TEMPSAC and TWLSQ require five hyperparameters to be specified: 1) the fixed forgetting factor λ, 2) the buffer length m, 3) the number of iterations of RANSAC k, 4) the number of initial points to fit the model n, 5) the threshold to define an inlier ε, and 6) the minimum required number of inliers z. The accuracy of the estimated platform motion from (8) can be evaluated using the relative distance (error) between it ($\hat{\mathbf{g}}_i$) and the known ground truth $\mathbf{g}_i$ at sample i. The error is calculated as $\hat{\mathbf{g}}_i \ominus \mathbf{g}_i$, where $\ominus$ is the inverse compounding operator that returns the relative pose as reported in [21]. For samples $i = 1,\dots,l$ where l is the desired sample to stop at, the optimization is formulated as a mixed integer programming problem to minimize the absolute pose error (*APE*) [22],

$$\begin{aligned} \text{min.} \quad & APE = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(\hat{\mathbf{g}}_i^w \ominus \mathbf{g}_i^w)^2} \\ \text{s. t.:} \quad & 0 \leq \lambda \leq 1 \qquad \varepsilon > 0 \\ & 1 \leq z \leq \min_i \{a_i\} \qquad n, z, k, m \in \mathbb{N}^+. \end{aligned} \tag{17}$$

TABLE I

COMPARISON OF AVERAGE ABSOLUTE POSE ERROR (*AAPE*) FOR DIFFERENT ENVIRONMENTS. THE *AAPE* IS PRESENTED IN METRES.

Algorithm	EC	IRL	EA
TEMPSAC	2.812	2.352	**5.615**
TWLSQ	**2.338**	**2.166**	5.653
KB	3.644	3.603	5.672

III. EXPERIMENTAL VALIDATION

The proposed methods are compared against the implementation introduced in [8] further referred to as KB, using the open source dataset Coloradar [18]. Coloradar is a multi-sensor dataset spanning six locations and over 13 kilometres. The dataset contains 3D measurements and motion. In this current paper, they are projected onto the xy-plane to be evaluated in 2D. For the purposes of our investigation, only single chip radar data (TI AWR1843BOOST-EVM), which has a sampling rate of 10 Hz is considered. The orientation of the platform ${}^{w}\phi_i$ is provided by an IMU.

Hyperparameter optimization: The data from 10 m of the Engineering centre hallway is used along with the corresponding ground truth. The hyperparameters are optimized using a genetic algorithm implemented with the *pymoo* Python package [23]. In total, it evaluated 2500 parameter combinations. The parameters that minimized (17) are, $\lambda = 0.815$, $m = 3$, $k = 1146$, $n = 2$, $\varepsilon = 0.0105$, and $z = 10$, resulting in $APE = 0.562$ m. These hyperparameters are used for all three methods, however, KB has no sliding window or weights.

Proposed methods validation: The indoor locations simulated are Edgar Classroom (EC), Intelligent Robotics Laboratory (IRL), and Edgar Army (EA), having a path length of 181 m, 81 m, and 131 m, respectively. Each location includes a variety of obstacles, from irregular walls to regular and repeating architecture. Since RANSAC is nondeterministic, each location was validated using 100 trials for each method.

Fig. 3, shows the xy-trajectory, *APE* over time and average absolute pose error *AAPE* (i.e., the sum of the *APE* for each location divided by the number of trials) for EC, IRL, and EA respectively. The standard RANSAC method (KB) performed the worst on average in all three locations. Taking a closer look at the xy-trajectory in IRL, the motion estimated by KB is less consistent than the other methods. For short periods it may perform marginally better than TEMPSAC or TWLSQ due to the lag introduced by the sliding window, but these small gains are negated by the large jumps in location seen in the upper right of IRL. These observations are confirmed in the APE-time plot, which shows large errors between 40 to 60 seconds. These errors are a result of mmWave measurements containing more outliers than inliers, causing the motion estimated to diverge from the platform's true motion. In contrast, TEMPSAC and TWLSQ did not experience these spikes in error at any location. The drop in EC *APE* at 145 seconds, is caused by the platform stopping.

These results show that by weighting the sliding window, TEMPSAC and TWLSQ are being constrained to the set of velocity model solutions consistent with previous measurements. The fact that our methods maintained a consistent trajectory in those locations supports our hypothesis that the inlier's Doppler velocity is consistent over short periods of time, while the outlier's measurement maintain no such temporal relationship. This is further evidenced by the fact that our methods have a lower *AAPE* with smaller interquartile range.

The *AAPE* for the 100 trials are shown in Table I. As expected from the results in Fig. 3, both our methods outperform KB, with TEMPSAC showing an average improvement of 19.5% and TWLSQ showing an average improvement of 25.3%. Additionally, TWLSQ outperforms TEMPSAC by 8%.

IV. DISCUSSION AND CONCLUSION

While mmWave sensors are more robust to environmental factors than other odometry sensors, measurements are subjected to high levels of outliers caused by multipath reflections. Yet, Doppler velocity ego-motion is traditionally estimated using data collected from a single time step [8], [10] thereby ignoring the time-dependency of successive measurements. In this paper, we proposed two methods that leverage the temporal relationship between measurement points to identify inliers using a temporally weighted sliding window. These methods improve motion estimation accuracy by 19.5% compared to traditional RANSAC based methods [8].

TWLSQ performed 8% better on average than TEMPSAC and 25.3% better than KB RANSAC. TWLSQ performed the best since it uniformly samples points over the entire sliding window. Uniformly sampling points ensures that the likelihood of generating a velocity model with specific parameter values is directly proportional to the density of points corresponding to those parameters. This means that TEMPSAC is more likely to generate a velocity model using inliers because they have higher densities due to being temporally consistent. Concurrently, the likelihood of generating a model using outliers is reduced since outliers in consecutive measurements have no temporal relation with one another. These findings are consistent with principles used in LIDAR scan matching, where the density of points is transformed into a probability distribution used for scan matching [16].

TEMPSAC performed 19.5% better than KB RANSAC. By using the sliding window, the density of inliers was improved like in TWLSQ. However, the likelihood of using these inliers was reduced due to its sampling method. By selecting points based on their temporal location within the buffer, the likelihood of generating a model based on a dense cluster of inliers is reduced, unless the cluster comes from the same measurement. This may explain why the genetic algorithm converged to having a buffer length of 3 and a forgetting factor near one; to make the probability of generating a model using inliers more evenly weighted across the sliding window. The improvements in motion estimation underscore the utility of leveraging temporal relationships in Doppler velocity measurements. This concept improves the accuracy of ego-motion estimation and paves the way for more robust and reliable applications of mmWave odometry.

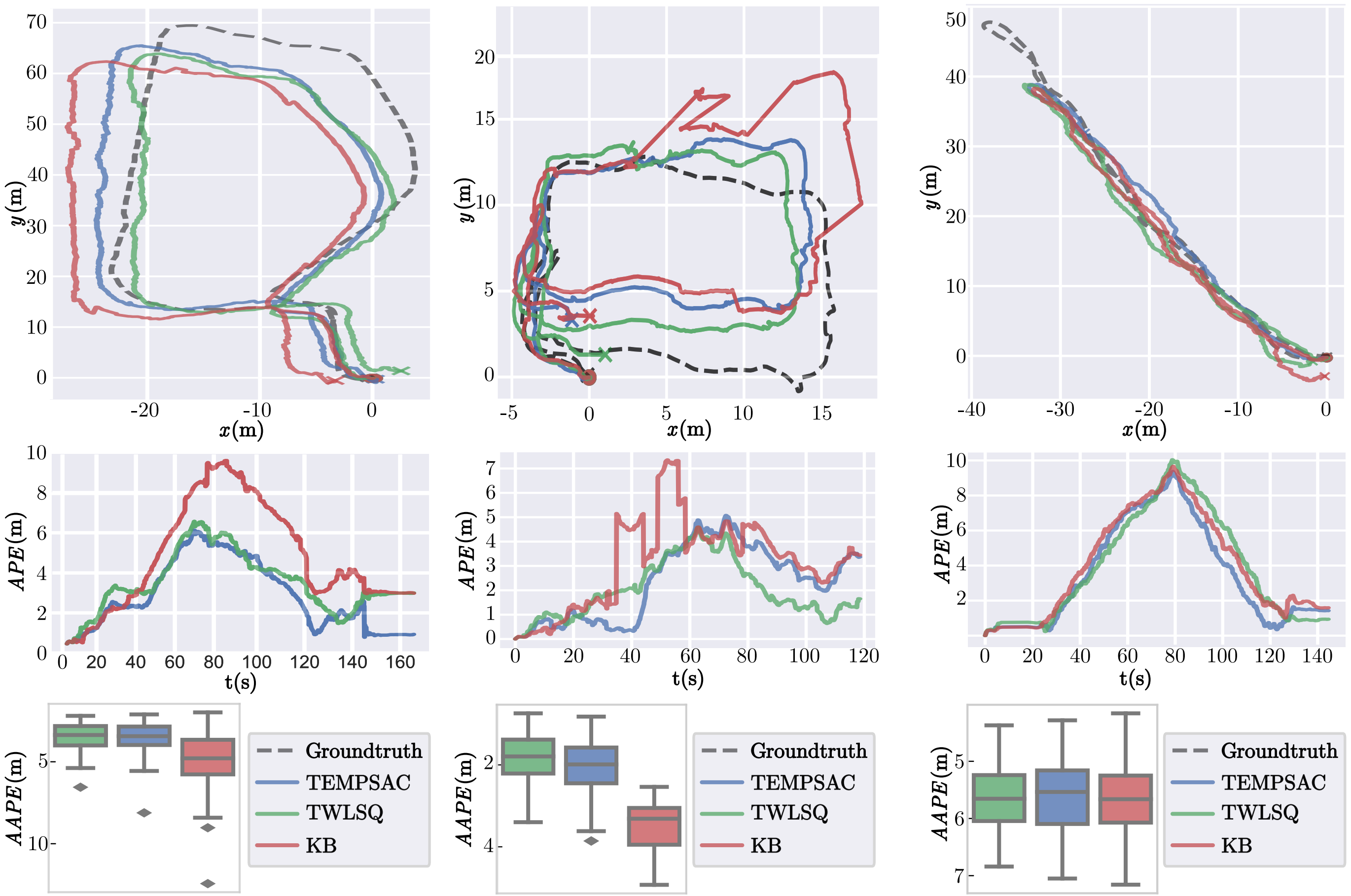

Fig. 3. A comparison of the results generated by TEMPSAC (blue), TWLSQ (green), and KB (red) over EC (left), IRL (centre), and EA (right) datasets. The top panel shows the xy trajectories, the middle panel shows the absolute pose error over time, and the bottom panel shows the average absolute pose error (*AAPE*) for the 100 trials. TEMPSAC and TWLSQ outperform KB.

References

[1] J. Leonard and H. Durrant, "Simultaneous map building and localization for an autonomous mobile robot," *Int. Conf. on Intelignet Robots and System*, pp. 1442–1447, 1991.

[2] K. Harlow et al, "A new wave in robotics: survey on recent mmwave radar applications in robotics," *arXiv preprint arXiv:2305.01135*, 2023.

[3] A. Kramer et al., "Radar-inertial ego-velocity estimation for visually degraded environments," in *Proc. ICRA*, 2020, pp. 5739–5746.

[4] X. Lu et al., "milliEgo: single-chip mmwave radar aided egomotion estimation via deep sensor fusion," in *18th Conf. on Embedded Networked Sensor Syst.*, 2020, pp. 109–122.

[5] C. Iovescu and S. Rao, "The fundamentals of millimeter wave sensors," *Texas Instruments*, pp. 1–8, 2017.

[6] C. Lu et al., "See through smoke: robust indoor mapping with low-cost mmwave radar," in *ACM MobiSys 2020*, 2020, pp. 14–27.

[7] Y. Almalioglu et al., "Milli-RIO: ego-motion estimation with low-cost mmwave radar," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3314–3323, 2020.

[8] D. Kellner et al., "Instantaneous ego-motion estimation using doppler radar," in *16th ITSC*, 2013, pp. 869–874.

[9] ——, "Instantaneous ego-motion estimation using multiple doppler radars," in *Inter. Conf. on Robotics and Aut.*, 2014, pp. 1592–1597.

[10] S. Kwon et al., "Radar sensor-based ego-motion estimation and indoor environment mapping," *IEEE Sensors J.*, 2023.

[11] M. Fischler and R. Bolles, "RANSAC: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[12] S.-W. Yang and C.-C. Wang, "Multiple-model RANSAC for ego-motion estimation in highly dynamic environments," in *Int. Conference on Robotics and Automation*, 2009, pp. 3531–3538.

[13] Y. S. Park et al., "3D ego-motion estimation using low-cost mmwave radars via radar velocity factor for pose-graph slam," *IEEE Rob. Automat. Lett.*, vol. 6, no. 4, pp. 7691–7698, 2021.

[14] Z. Zeng et al., "Joint velocity ambiguity resolution and ego-motion estimation method for mmwave radar," *IEEE Rob. Automat. Lett.*, 2023.

[15] S. Kwon et al., "Radar sensor-based ego-motion estimation and indoor environment mapping," *IEEE Sensors J.*, 2023.

[16] S. Lovett et al., "Level plane slam: Out-of-plane motion compensation in a globally stabilized coordinate frame for 2D SLAM," in *Int. Conference on Systems Man and Cybernetics*, 2023, pp. 3355–3360.

[17] N. Iqbal et al., "Multipath cluster fading statistics and modeling in millimeter-wave radio channels," *IEEE Trans. Ant. Propgat.*, vol. 67, no. 4, pp. 2622–2632, 2019.

[18] A. Kramer et al, "Coloradar: The direct 3D millimeter wave radar dataset," *The Int. J. of Robot. Res.*, vol. 41, no. 4, pp. 351–360, 2022.

[19] M. Waine et al., "3D shape visualization of curved needles in tissue from 2D ultrasound images using ransac," in *Proc. ICRA*, 2015, pp. 4723–4728.

[20] D. Bodenham et al., "Continuous monitoring for changepoints in data streams using adaptive estimation," *Statist. and Comput.*, vol. 27, no. 5, pp. 1257–1270, Sep 2017.

[21] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Automat. robots*, vol. 4, pp. 333–349, 1997.

[22] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Intl. Conf. on Intelligent Robots and Syst.*, 2018, pp. 7244–7251.

[23] J. Blank and K. Deb, "pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.