Multi-Sensor Fusion for Real-Time Object Tracking

Sakshi Verma[1] and Vishal K. Singh[1*]

[1*]Wireless Communication and Analytics Research Lab (WCARL), Department of Computer Science, Indian Institute of Information Technology, CG City, Lucknow, 226002, Uttar Pradesh,India.

*Corresponding author(s). E-mail(s): vashukrishna@gmail.com; Contributing authors: sakshiv1357@gmail.com;

Abstract

Reviewer 2: Comment 3

Accurate orientation and position estimation are critical elements in optimizing real-time object tracking performance when leveraging smartphone sensors such as accelerometers and gyroscopes. The primary challenges encountered in smartphone-based object tracking are attributed to the GPS signal, canyon effect, and orientation errors, accumulation error in sensor. To address these limitations, a novel approach is proposed wherein a smartphone application is developed based on IMU Multi -sensor fusion using Kalman filter and Rotation vector. The proposed approach integrates Kalman filtering to fuse sensor data and leverages the rotation vector for precise orientation estimation. Additionally, geohash filtering is employed to efficiently proficiency in quantifying intricate spatial interdependencies and display track paths on maps within the application. A detailed mathematical analysis and thorough comparison with existing algorithms in the field proves the dexterity of the proposed object tracking scheme. The comprehensive evaluation showcases the algorithm's capability and advancement compared to state-of-the-art approaches. .

Keywords: Kalman Filter, Rotation Vector, Geohash Filter, Linear Acceleration

1 Introduction

Object tracking has emerged as an interesting application which involves real time data classification and interface for highly accurate inference [1]. Practical implementation of tracking algorithms, often involves the use of smartphone sensors such as GPS and Gyroscope. However, challenges imposed due to accumulated sensors error, environmental components, location error in smartphone devices, result in steep degradation in the overall accuracy and performance of the existing methods.The pressing challenges in object tracking applications may be summarised as follows.

1. Accuracy: Object tracking accuracy depends on the accuracy of the accelerometer's calibration. However, signal noise and biases reduce the calibration accuracy and algorithm reliability.
2. Object Localization: IMU can be used for object tracking in outdoor localization. However various factors such as external electromagnetic noise or sensor drifts impact the accuracy of the IMU based localization system.

Existing methods of object tracking employ several mechanisms for addressing these issues.For instance the authors in [2] used the multivariate data fusion (MVDF) technique for connected ITS vehicles.In the realm of intelligent transportation systems (ITS),

Reviewer 2: Comment 4

The prediction of urban traffic crowd flows holds significant importance. However, effectively capturing the intricate spatial connections between different regions and dynamic temporal relationships across various time periods presents a challenge. To address this obstacle, a novel solution called DHSTNet is introduced. DHSTNet is a dynamic deep hybrid spatio-temporal neural network specifically designed for achieving highly accurate predictions of citywide traffic crowd flows. It incorporates four key properties, namely closeness volume, daily volume, trend volume, and an external branch, enabling precise traffic flow predictions in every region of a city [3],[4]. The authors utilized regression-based computation and filtration process to reduce the errors in the fusion process.While, Network simulator experiments were used to measure model performance. Object tracking in dynamic environments is a big challenge for researchers as it involves information about object monitoring which involve various parameters [5].

Reviewer 2: Comment 4

Wearable sensors offer a another promising approach for object tracking by leveraging the capabilities of these sensors. In this approach a feature fusion technique is applied, utilizing covariance matrices to extract correlation information from accelerometer-based inertial sensor fusion. The findings demonstrate that the Affine-Invariant kernel achieves the highest accuracy when used with both RBF and sigmoid activation functions [6].The author in [7] proposed a real time tracking application based on camera and IMU sensors data which the utilized visual-inertial odometry (VIO) to integrate visual and IMU data. IMUs are portable devices that are used for capturing human movement. IMUs are also used for tracking objects in various positions [8]. However, the orientation of IMUs is the biggest challenge faced during object tracking. Some studies have used the Extended Kalman Filter (EKF) to overcome this problem. The main utility of the EKF algorithm is to find the position of the moving object in a dynamic environment. Similarly, Error-State Kalman Filter (ESKF) was used for estimate the Inertial measurement units (IMU) orientation. Based on the experimental results, the author showed that ESKF provides higher accuracy than coordinated measure machine (CMM). Precise localization is vital for accurate smartphone-based navigation in the real world. To enhance localization accuracy, a fingerprint-based approach is proposed, drawing inspiration from microbat echolocation. This method incorporates a Bayesian-rule based objective function for improved smartphone-based navigation systems[9] Kalman Filter can also be used with a network of LIDAR sensor to track vehicles in real-time under parking like conditions. The authors in [12] used RANSAC algorithm, for achieving a mean lateral error of 6.3 cm and a longitudinal error of 8.5 cm. Some authors have used Android Studio and GPS to create location tracking application.GPS systems are used for real-time tracking and incident alert system in commercial city buses [13]. Similarly, the GPS can be used for creating a specific application for location tracking and route identification. However, Canyon effect and limitation of satellite visibility in dense urban environments impact the positioning estimation and GPS tracking. As such, GPS is used for acquire location, Open Street Map for layout, and Android studio for development [14]. Many IoT cloud systems for traffic monitoring and accident prevention use Mobile sensors such as GPS/GPRS to collect Geo-location and speed data. These cloud-based systems use various other tools such as OpenStreetMap for visualization and MongoDB for data management and provide real-time traffic movement data [15] resulting in improved detection accuracy and computational efficiency. The main focus of this research is to develop a cost-effective object tracking solution that utilizes phone sensors, specifically without relying on audio and video data. The objective is to improve the accuracy of the tracking process by leveraging the capabilities of phone sensors while keeping the solution affordable and accessible. Mobile sensors such as Gyroscope, Accelerometer, and GPS are used for input data collection in real-time tracking of vehicles, including positioning and manoeuvring of vehicles. Filters such as Exponential Moving Average Filter, Low pass filter and Kalman filter were used for fusion and smoothing of data [10].IMU integrates with GPS and a known map along with driving lanes and road markers detected[16]

Reviewer 1: Comment 1 Reviewer2 : Comment 1

The proposed work is therefore aimed at achieving improved accuracy and reduced localization error by making use of sensor fusion of smartphone sensors. Kalman filter along with Geohash filter is used to develop an algorithm which requires minimum external hardware (only smartphone and laptop), thereby enhancing its affordability, efficiency, portability and maintainability. Additionally, the use of Kalman and Geo hash filter is aimed at improving the overall accuracy of the proposed method.

The contributions of the proposed work are summarised as follows:-

1. A real time object tracking algorithm is proposed to address the issues imposed due to sensor drifts, GPS signal lost and accelerometer's calibration, orientation error.

2. A detailed mathematical analysis to prove the relation between the rotation matrix and axis-angle for orientation parametrization.
3. An extensive comparative analysis of the results obtained for the accuracy of the proposed method is also presented,

Rest of paper is organized as follows. Section 2 discusses the problem description related to real-time object tracking. Section 3 presents the network parameter and framework. Section 4 describes the system model of the application while section 5 discusses the proposed architecture. Section 6 presents the analytical analysis of the proposed method.In section 7, the simulation setup is discussed and finally, the paper is concluded in Section 8 and the future work is discussed in section 9.

2 Problem Description

Existing methods of object tracking through motion transformation ,work on the general straightforward idea of transforming the motion information i.e orientation, from a smartphone to that of the object i.e vehicle.The general process of object's orientation to position is shown in figure 1. The acceleration data from the phone is transformed to vehicle's acceleration in several steps .These steps include obtaining the gravity direction with the help of mobile orientation, followed by deduction of acceleration on a horizontal plane. Finally the transformation matrix is obtained by a set of mathematical computations. The matrix,thus obtained, converts the phone's acceleration data to that of the vehicle. However, a careful observation of the facts, reveal that the accuracy in such approaches suffer significantly because of the following reasons:

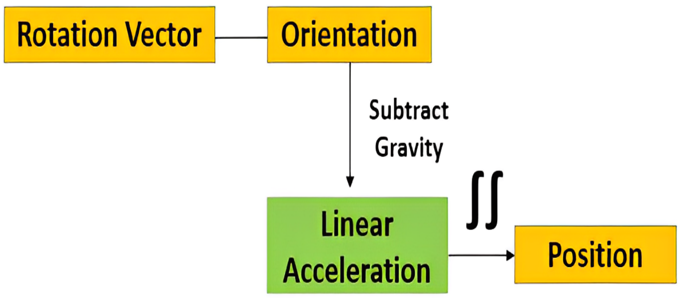

Fig. 1 Object's Orientation to Position

1. **Vehicle on slope :-** In case where the vehicle is on slope, the direction of the gravity is considered to be Z axis of the vehicle , which is not necessarily true.
2. **Noise :-**The readings of accelerometer are noisy because of the frequently varying terrain and driving conditions. This negates the general assumptions of maximum horizontal acceleration to be always on Y axis.
3. **Reliability:-** The reliability of the direction of maximum horizontal acceleration is significantly affected with the continuously changing phone position due to rough terrain, driving dynamics etc.

Table 1 Symbols With Description

Symbols	Description	Symbols	Description	Symbols	Description
A	Raw Acceleration	$\Delta\phi$	Latitude1-Latitude2	V_t	Measurement Noise
t	Discreet time slots	P	Co-variance matrix	n	Unit vector
G	Gravity	w	Noise due to uncertainty	H	Observational matrix
x	State of an object	x_j, y_j	Estimated position	R	Covariance Observation Noise
u	Control vector matrix	k_1, k_2, k_3	Components of the vector	N	no of iteration
B	Control matrix at time t	z	Measurements data from sensors	Q_t	Co-variance process noise
$\Delta\sigma$	Interior spherical angle	F	State transition matrix	V	noise due to measurement
X_t	State vector matrix	x_a, y_a	Actual position	K_t	Kalman Gain

3 Network Parameters and framework

3.1 Linear Acceleration

Linear acceleration is used for estimation of position and speed of the device, and is given by:

$$A_L = A - G \tag{1}$$

While G is obtained using the orientation of a device.

$$Velocity = \int_0^t A dt \tag{2}$$

$$Position = \int_0^t \int_0^t A dt \tag{3}$$

3.2 Rotation Vector

A rotation vector is a synthetic sensor that provides the device's orientation with the help of an accelerometer, magnetometer,and Gyroscope (optional). The orientation of the device can be represented using quaternion, Euler angles, and rotation matrix. The rotation vector gives the orientation of the device in the form of a quaternion. A quaternion is a four-element vector that also shows the amount of rotation. Euler angle represents 3D orientation in the form of roll angle, pitch angle, and yaw angle with respect to X, Y Z direction.It is important to note that quaternions are difficult to visualize, while the Euler angle suffers from the singularity at transitions of quadrants,leading to gimbal lock. Thus, a rotation vector is more suitable to represent the orientation. The Figure 2 illustrates the coordinate system used by the rotation vector sensor and

Reviewer 3: Comment 1

the table 1 shows the symbols along with their descriptions as used throughout this work.

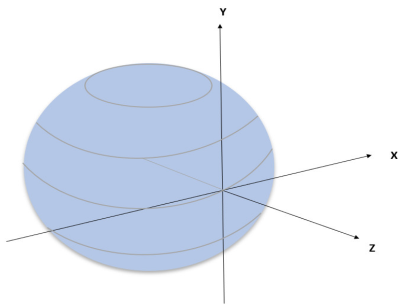

Fig. 2 Rotation Vector Sensor with Respect to Earth

4 System Model

The physical motion and measurement model of an object is given by :

$$X[t] = f(x[t-1], u[t], w[t]) \tag{4}$$

$$z[t] = g(x[t], v[t]) \tag{5}$$

The equations f and g represent the abstract models derived from the underlying physical dynamics and inherent properties of the sensing devices. These models encapsulate the relationship between the state and measurement variables. Generally, object is tracked through one or more IMU sensors such as GPS, accelerometer, rotation vector.Each sensor obtains its own estimate which is integrated through the fusion process.

4.1 System State Model

An object can be tracked in 2-Dimensional using X and Y coordinates. The system state transfer function is given by:

$$X_t = TX_{t-1} + BU_t + W_t \tag{6}$$

$$T = \begin{bmatrix} 1.0 & 0.0 & dt & 0.0 \\ 0.0 & 1.0 & 0.0 & dt \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad X = \begin{bmatrix} X \\ Y \\ XVelocity \\ YVelocity \end{bmatrix}$$

$$B = \begin{bmatrix} dt^2 & 0.0 \\ 0.0 & dt^2 \\ dt & 0.0 \\ 0.0 & dt \end{bmatrix} \quad U = \begin{bmatrix} X_Acceleration \\ Y_Acceleration \end{bmatrix} \quad W = \begin{bmatrix} W_{xi} \\ W\prime_{xi} \\ W_{yi} \\ W\prime_{yi} \end{bmatrix}$$

4.2 Measurement Model

At a given time "t" the measurement z_t of the state estimate x_t is calculated by following equation

$$Z_t = H_t.x_t + v_t \tag{7}$$

Where, H_t maps the true state space into measured space.

5 Proposed Architecture

A detailed structure of object tracking using IMU sensors and Kalman filter is shown in figure 3.The proposed architecture is divided into two main stages:

1. **Stage 1:-**

   Sensor fusion with GPS and accelerometer

   Position and speed of an object is determined by sensor fusion between GPS and accelerometer.

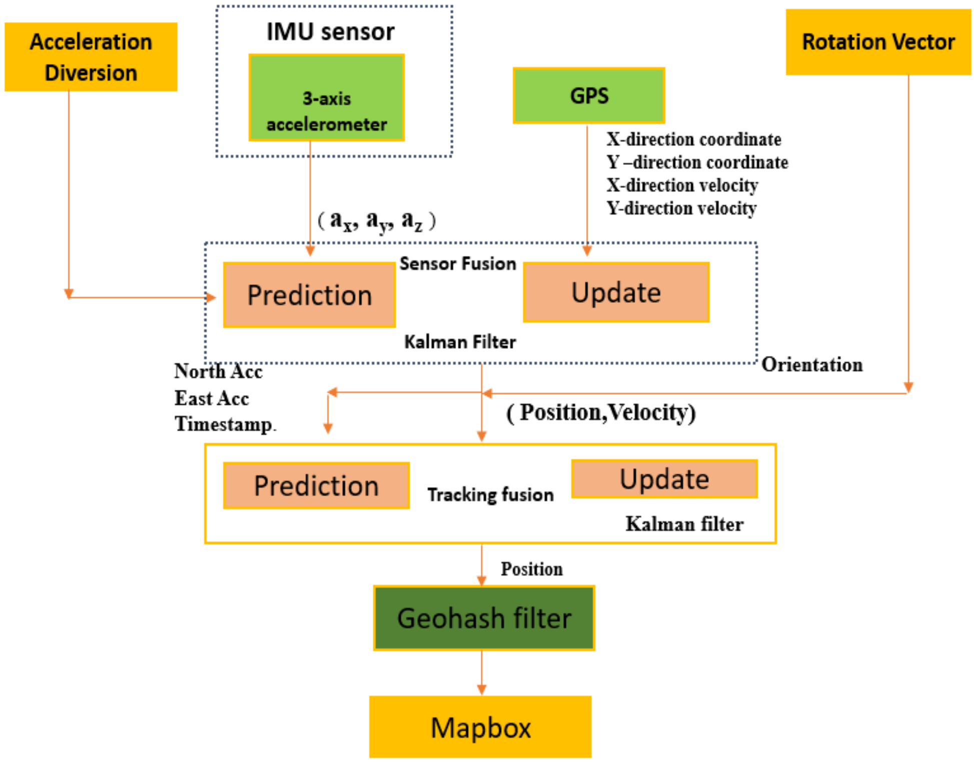

Fig. 3 Proposed Architecture

2. **Stage 2:-**

   Tracking fusion using Kalman filter and Geo hash filter

   This is used to determines the location of the object.

Reviewer 1: Comment 2

5.1 Sensor fusion with GPS and accelerometer

Sensor fusion between GPS and accelerometer is utilized to measure speed and position of any object. As shown in figure 3, input values from GPS and accelerometer sensor goes through the Kalman filter for sensor fusion.

Reviewer 2: Comment 2

The Kalman filter relies on accurate error estimation of measurements, which poses a challenge when utilizing sources like smartphones or GPS updates that lack inherent error estimates for speeds [17]. Therefore, a method was developed to estimate the error in GPS speed updates. While GPS speed updates maintain accuracy during constant vehicle speed, they introduce deviations once acceleration is applied. Before fusion, acceleration deviation is calculated for the Covariance process error matrix (Q_t) and Covarience observtaion noise matrix (R). This matrix is used in the Kalman filter fusion during predict and update stages. Acceleration deviation which is represented by σ,is calculated based on the following formula.

$$\alpha = \sqrt{[\sum_{n=1}^{\infty} \frac{(x-\overline{x})^2}{N})} \tag{8}$$

The mean value of measurement calibrations used in the formula is calculated by the acceleration value. Kalman filter is an iterative process or data fusion algorithm that works in two stages namely: Prediction and updation that provides estimates of unknown variables given the measurements observed over time. Updated values from the sensor are compared with the predicted value and based on that information, object's speed and position is updated in the system through following steps:-

STEP 1 Prediction. Prediction is a first step in Kalman filter.The predict states updates the last estimation using the propagation model and updates the co-variance accordingly. X- direction acceleration and Y- direction acceleration input values from accelerometer sensor act as predict values. Following equation represents the prediction stage:

$$X_{t|t-1} = FX_{t-1|t-1} + BU_t \tag{9}$$

x_t is the state vector containing the terms of interest for the system (e.g., position, velocity, heading) at time tu_t is the vector containing any control inputs (steering angle, throttle setting, braking force) $F_t$is the state transition matrix which applies the effect of each system state parameter at time t-1 on the system state at time t (e.g., the position and velocity at time t-1 both affect the position at time t) B_t is the control input matrix which applies the effect of each control input parameter in the vector u_t on the state vector (e.g., applies the effect of the throttle setting on the system velocity and position) $u_t$is the vector containing the process noise terms for each parameter in the state vector. The process noise is assumed to be drawn from a zero mean multivariate normal distribution with covariance given by the covariance matrix Q_t. Sensors provide the input values to track the position and velocity of the object. After receiving the input values from the sensor, Kalman filter estimates the values.

1. Firstly, GPS provides the measurements to the sensor fusion method for position and speed of an object. The measurements are passed through predict step of Kalman filter to determine the state estimates of an object at time t.
2. Previously determined acceleration deviation acts as σ value in the co-variance matrix for the fusion algorithm. Based on the σ, the process noise of the co-variance matrix Q_t is calculated.

$$F = \begin{bmatrix} 1.0 & 0.0 & dt & 0.0 \\ 0.0 & 1.0 & 0.0 & dt \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} X = \begin{bmatrix} X \\ Y \\ XVelocity \\ YVelocity \end{bmatrix}$$

$$B = \begin{bmatrix} dt^2 & 0.0 \\ 0.0 & dt^2 \\ dt & 0.0 \\ 0.0 & dt \end{bmatrix} U = \begin{bmatrix} X_Acceleration \\ Y_Acceleration \end{bmatrix}$$

$$P_{t|t-1} = FP_{t-1|t-1}F^T + Q_t \tag{10}$$

Covariance of Process Noise

$$Q_t = \begin{bmatrix} \sigma^2 pos_x & 0.0 & \sigma.pos_x & \sigma.vel_x & 0.0 \\ 0.0 & \sigma^2 pos_y & 0.0 & \sigma.pos_y & \sigma.vel_y \\ 0.0 & 0.0 & \sigma^2 vel_x & 0.0 & \\ 0.0 & 0.0 & 0.0 & \sigma^2 vel_y & \end{bmatrix}$$

STEP 2. Updation.

1. This stage updates the object position based on the prediction and sensor reading (measurements). GPS input values: X- direction coordinates, Y- direction coordinates, X- direction velocity and Y- direction velocity act as update values. Updation stage starts with the calculation of Kalman gain.

   The Kalman gain represents the weighting assigned to both the measurements and the current-state estimate, offering the ability to fine-tune performance according to specific objectives. A higher gain emphasizes recent measurements, leading to a more responsive adaptation to them. Conversely, a lower gain prioritizes adherence to model predictions. When the gain is close to one, the estimated trajectory becomes more erratic, whereas a gain close to zero minimizes noise but reduces responsiveness, resulting in a smoother trajectory. Kalman gain K_t factor can be calculated using the following equation:

$$K_t = P_{t|t-1}H^T(HP_{t|t-1}H^T + R)^{-1} \tag{11}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_o = \begin{bmatrix} \sigma^2.pos & 0.0 & 0.0 & 0.0 \\ 0.0 & \sigma^2.pos & 0.0 & 0.0 \\ 0.0 & 0.0 & \sigma^2.vel & 0.0 \\ 0.0 & 0.0 & 0.0 & \sigma^2 vel \end{bmatrix}$$

2. After calculation of Kalman gain, updated state estimate X_t is calculated using the following equation.

$$X_{t|t-1} = X_{t|t-1} + K_t(Z_t - HX_{t|t-1}) \tag{12}$$

Where,

$$Z = \begin{bmatrix} X \\ Y \\ XVel \\ YVel \end{bmatrix}$$

STEP 3. Error Estimation. Next step involves estimation of error co-variance using the following equation.

$$P_{t|t} = (I - K_t.H)P_{t|t-1} \tag{13}$$

Reviewer 1: Comment 2

5.2 Tracking fusion using Kalman filter and Geo hash filter:

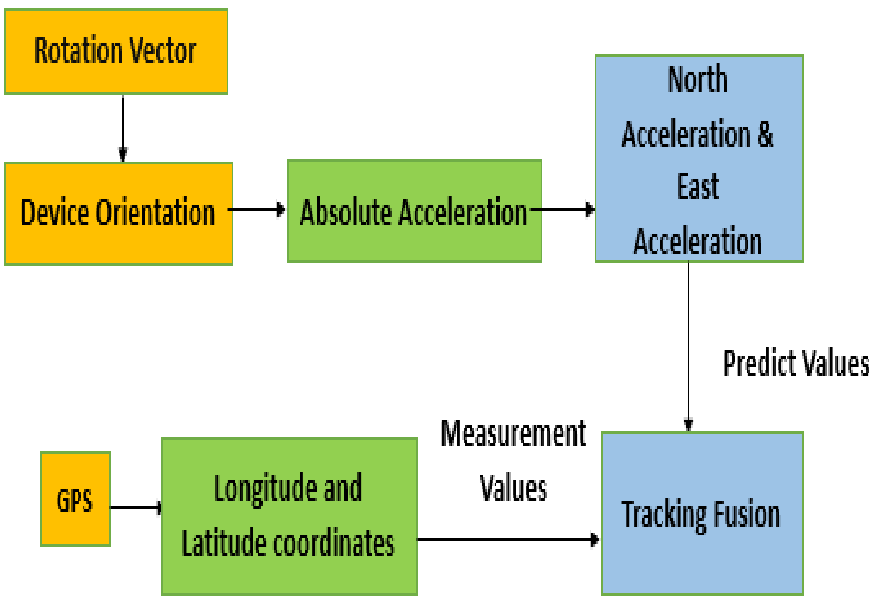

Fig. 4 Tracking Fusion

Output obtained in the previous fusion (Position and Velocity in x and y direction) is again fused with kalman filter for object's location.After that Geo hash filter used object's location for converting the longitude and latitude coordinates into string or code that gives speed to search spatial information [18].During tracking-fusion, geohash filter gives more precise value for location on the map. In the context of tracking fusion, the north and east accelerations serve as predictive components, contributing to the overall prediction stage. Determining the values of north and east acceleration entails the computation of absolute acceleration, which is obtained through the consideration of device orientation. Following steps are followed in this process:

Calculation of absolute acceleration value based on the device's orientaion.

1. Absolute acceleration is calculated based on the orientation of device with the help of rotation vector. The analytical analysis section 6 of this paper presents the demonstrated working model of rotation vector. Main work of the rotation vector is to reduce some of the complexity, which is created due to combination of multiple sensors used together. Thus, it is often preferable to use the rotation matrix instead of the accelerometer and magnetometer to determine object orientation.
2. After finding the object orientation, based on the above steps, absolute acceleration is determined towards east and north direction.
3. North acceleration refers to the absolute acceleration in north direction excluding gravity, while East acceleration refers to the absolute acceleration in east direction excluding gravity.

   Reviewer 1: Comment 2

4. As depicted in Figure 4, the presented methodology involves the utilization of north and east accelerations, along with timestamp information, as predictive states. The GPS longitude and latitude coordinates are considered as measurement values. The data is processed using the Kalman fusion algorithm, following previous stages. By applying the equations of the Kalman filter, the desired outcome of object locations is obtained.
5. The main purpose of tracking fusion is filtering the GPS location data and add more parameters such as acceleration deviation, GPS min time, GPS min distance, geohash precision default sensor frequency, position factor, and velocity factor to get more precision and accuracy in localization.
6. Addition of more parameters and application of geohash filter on the values obtained through Kalman filter, provides more precise and accurate results.

Initially, the value of various parameters such as acceleration deviation, GPS min time, GPS min distance, sensors frequency, Geohash precision and Geohash min-point are pre defined. Later, in each iteration, the measurement values, as obtained through GPS, are considered. The information thus obtained, i.e. the measurement value of GPS, provides the measurement of object's own state. Finally the predication and update state procedure are conducted to estimate the state of the tracked object. Based on the output in this step, speed, bearing, altitude, longitude and latitude, position error and velocity error are calculated. Based on the location co-ordinates, the distance between two points can be calculated using the formula as given in equation

Algorithm 1 :Proposed Algorithm

Algorithm
Initialize: Acceleration deviation, GPS min time, GPS min distance, GeoHash precision, GeoHash min point
For t = 1 : T **do**
Receive : The Measurement value from GPS as Z Matrix and predication value from accelerometer sensor

Filtering:
Prediction Step :

$$X_{t|t-1} = FX_{t-1|t-1} + BU_t$$
$$P_{t|t-1} = FP_{t-1|t-1}F^T + Q_t$$

Kalman Gain

$$K_t = P_{t|t-1}H^T(HP_{t|t-1}H^T + R)^{-1}$$

Update Step

$$X_{t|t-1} = X_{t|t-1} + K_t(Z_t - HX_{t|t-1})$$
$$P_{t|t} = (I - K_t.H)P_{t|t-1}$$

End For

14. This work uses mapbox for the visualization of location and path.

$$\Delta\sigma = 2arc\sin\sqrt{(\sin(\frac{\Delta\phi}{2}) + \cos\phi_s\cos\phi_f\sin(\frac{\Delta\lambda}{2})} \tag{14}$$

6 ANALYTICAL ANALYSIS

Orientation of an object plays an important role in object tracking. Fields such as spacecraft, vehicle tracking use various sensors such as gyrosocpe, accelerometer magnetometer for estimation orientation. There are many approaches for orientation parametrization which includes use of quaternion, Euler angles, and rotation matrix. However, different estimation methods and parametrization of the orientation limit the applicability.

6.1 Parametrizing Orientation

When a vector is rotated in R^3, It only changes its direction while length remain constant and group of rotaion in R^3 is known as Special Orthogonal group. SO(3).

6.1.1 Rotation Matrix

Rotation of a vector in different R^3 Rotation Matrix represents by the following equation:

$$R.R^K = R^K.R = I_3, \qquad detR = 1 \tag{15}$$

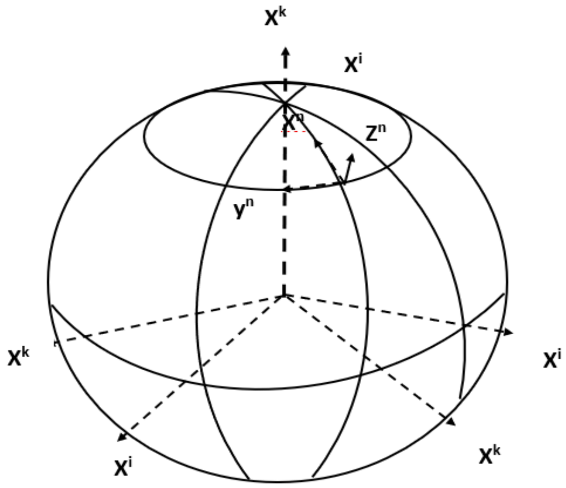

Fig. 5 Rotation of Vectors to Different Coordinate Frames

Let us considers two co-ordinate frame denoted by i and k as shown in figure 5. a vector x in the i-frame rotated to the k-frame as

$$x^k = R^{ik}.x^i \tag{16}$$

A rotation matrix is one of the Parametrizing orientation of an object it contains 9 components which depend on each other as defined in equation 15.

6.1.2 Rotation Vector

According to the Euler rotation theorem, a single rotation around a fixed point is equivalent to a single rotation through the axis which runs through the fixed point. An angle α and unit vectors n can be utilized to express the rotation between two frames. In figure 6, a vector x rotated an angle β around the unit vector n. Let us suppose that vector x^i is rotated in i coordinate frame is express as x. Vector x decomposed into a parallel to the axis n and its express as $x_{\|}$ and orthogonal to denoted by $x_!$ as.

$$x^i = x^i_{\|} + x^i_{\perp} \tag{17}$$

Using the geometric reasoning we can infer that

$$x^i_{\|} = (x^i.n^i)n^i, \tag{18}$$

Where "." shows the inner product.

$$x^i. = (x^i._{\|}) + (x^i._{\perp}) \tag{19}$$

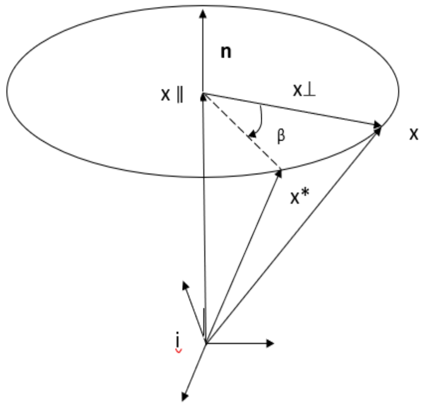

Fig. 6 Rotation of Vector x by an Angle α and Around Unit Vector n

where,

$$(x^i.)_{\|} = x^i \parallel . \tag{20}$$

$$(x^i.)_{\perp} = x^i_{\perp} \cos\beta + (x^i \times n^i)\sin\beta. \tag{21}$$

Hence, $x^i.$ can be expressed in terms of x^kas

$$x^i. = (x^i.n^i)n^i + (x^i - (x^i.n^i).n^i)\cos\beta + (x^i \times n^k)\sin\beta$$

$$x^i. = x^i\cos\beta + (x^i.n^i).n^i)(1-\cos\beta) - (x^i \times n^i)\sin\beta \tag{22}$$

Using the equivalence between $x^i.$ and x^i, this implies that

$$x^k. = x^i\cos\beta + .n^i(x^i.n^i)(1-\cos\beta) - (x^i \times n^i)\sin\beta \tag{23}$$

This equation is known as Euler formula or rotation formula. To express the equivalence between equation (23)and rotation matrix parametrization. Here, to show the fact that a cross product written as matrix vector product.

$$k \times i = [k\times]i = -[i\times]k, [k\times] = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \tag{24}$$

In addition, given vectors k, i and multiple cross product can be shown in terms of the inner product as

$$k \times (i \times w) = i(w \times k) - w(ki) \tag{25}$$

using equation (23) can be rewritten as

$$x^k = x^i\cos\beta + .n^i(x^i.n^i)(1-\cos\beta) - (x^i \times n^i)\sin\beta$$

$$x^k = x^i\cos\beta + (n^i \times x^i) + x^i(1-\cos\beta) - (n^i \times x^i)\sin\beta$$

$$x^k = (I_3 - \sin\beta[n^i\times] + (1-\cos\beta)[n^i\times]^2)x^i \tag{26}$$

Differentiated between equation (16) and (26) it clearly shows that a rotation martix can be parametrized in terms of β and n as

$$R^{ki}(n^i,\beta) = (I_3 - \sin\beta[n^i\times] + (1-\cos\beta)[n^i\times]^2 \tag{27}$$

$$R^{ki}(n^i,\beta) = exp(-\beta[n^i\times]) \tag{28}$$

$$\begin{aligned} exp(-\beta[n^i\times]) &= \sum_{t=0}^{\infty}\frac{1}{t!}(-\beta[n^i\times])^t \\ = (I_3 - [n^i\times] + \frac{1}{2!}(\beta^2[n^i\times])^2 &+ \frac{1}{3!}(\beta^3[n^i\times]) - \frac{1}{4!}(\beta^4[n^i\times])^2 - .... \\ = (I_3 - (\beta - \frac{1}{3!}\beta^3 + ...)[n^i\times] &+ (\frac{1}{2!}\beta^2 - \frac{1}{4!}\beta^4 + ...)[n^i\times]^2 \\ = (I_3 - \sin\beta[n^i\times] &+ (1-\cos\beta)[n^i\times]^2 \end{aligned} \tag{29}$$

As shown equation (27) and (28). the rotation matrix can be directly expressed in terms of axis-angle representation.

7 Simulation Setup

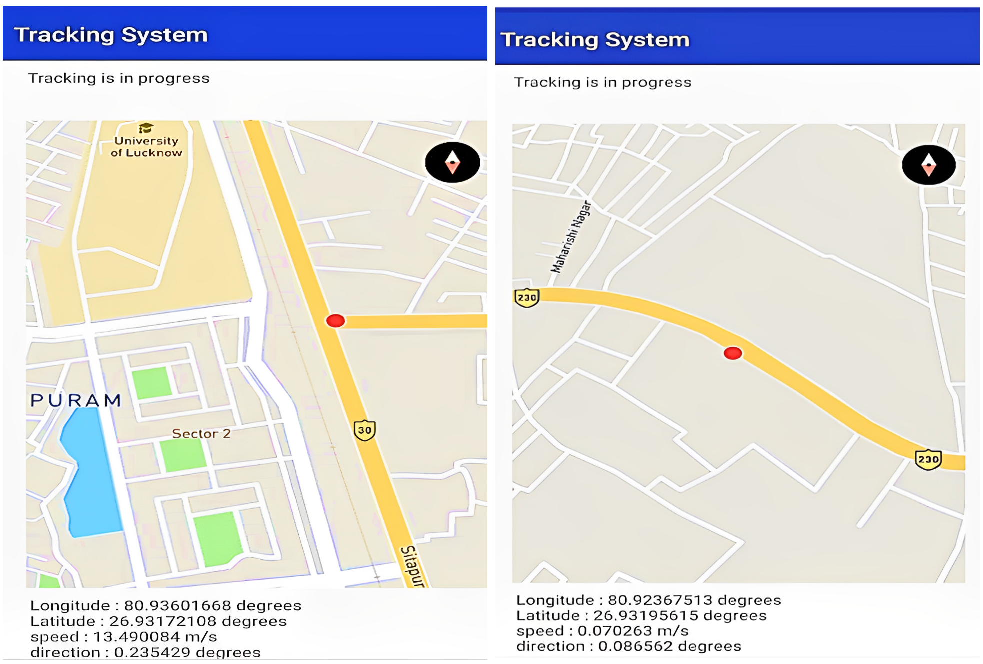

Fig. 7 User Interface During Turning and Stable Condition of Object

A Samsung M-21 mobile phone was used to run the proposed object tracking application. The details of the processing unit and related software's is shown in table 2. The application was used for tracking a four-wheeler on a 6 KM route with three different GPS frequencies, specifically at 0.4 Hz, 0.2 Hz and 0.033 Hz. The route had quiet common characteristics such as long straights, sharp turns, long smooth turns, short straights etc, which are suited for testing the performance of the application. A model scenario of the tracking area is shown in figure 8. Along with the user interface (UI) of the proposed application is shown in figure 7.

Fig. 8 Model View of Tracking

Table 2 System Components

Components	Description
Software	Android Studio 4.0
Processor	Windows 10
Mobile Device	Samsung M-21
Vehicle	4-wheeler

Reviewer 1: Comment 3

7.1 Performance Analysis

The performance of the proposed method is tested under three different scenarios by obtaining the error measurements for each scenario:
1. Scenario 1: GPS frequency - 0.4 HZ
2. Scenario 2: GPS frequency - 0.2 Hz
3. Scenario 3: GPS frequency - 0.033 Hz

The corresponding results for the GPS error and the proposed approach, are presented below:

7.1.1 Scenario 1: GPS frequency - 0.4 HZ

The figure 9 shows the path testing route for the proposed approach with GPS frequency 0.4 Hz. The corresponding error measurements, obtained with mobile GPS and that with the proposed approach for varying time duration, is presented in figure 10. As evident from the reported observations, the proposed approach is able to achieve significantly better results as compared to the mobile GPS owing to the improved tracking due to double filtration and optimized rotation vector calculations. The rotation matrix, obtained with the proposed approach, allows reduction in the signal noise and improved prediction of gravity, in cases where the vehicle is on slope. A steep peak is reported at around 03 seconds and 3.6 seconds of the test, which may be because of sudden change in the driving dynamics in real world scenario.

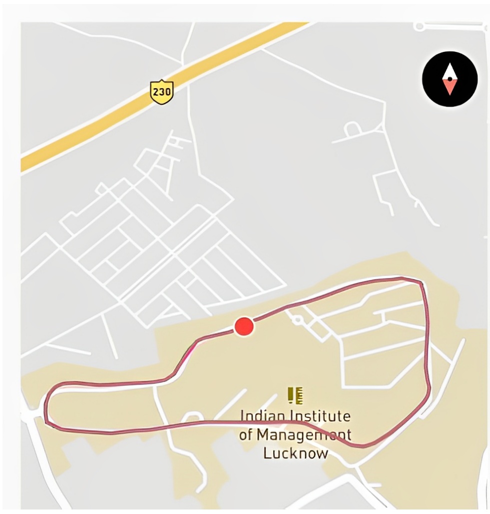

Fig. 9 Path Testing

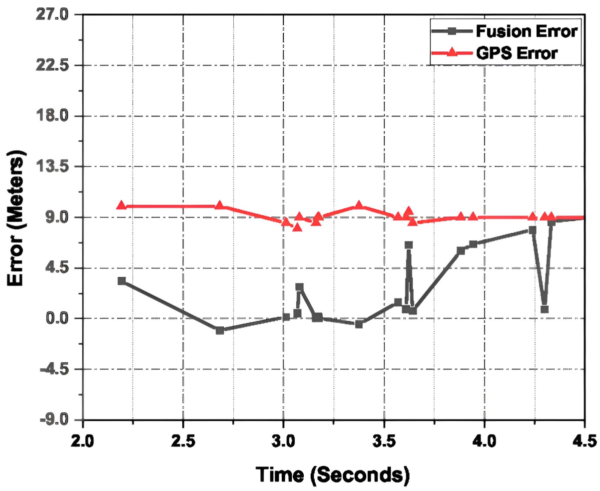

Fig. 10 Evaluation with .4HZ

7.1.2 Scenario 2: GPS frequency - 0.2 HZ

The observations for 0.2 Hz GPS frequency are reported in figure 11 and figure 12, where the proposed method outperforms the GPS based tracking by a significant margin. The proposed approach reports higher error between 06 seconds and 08 seconds, however, it quickly recovers and improved results are reported at 10 seconds and onwards. A prominent reason for the sudden increase in the error values and recovery thereafter, may be accounted for as the observations are reported for an average of 50

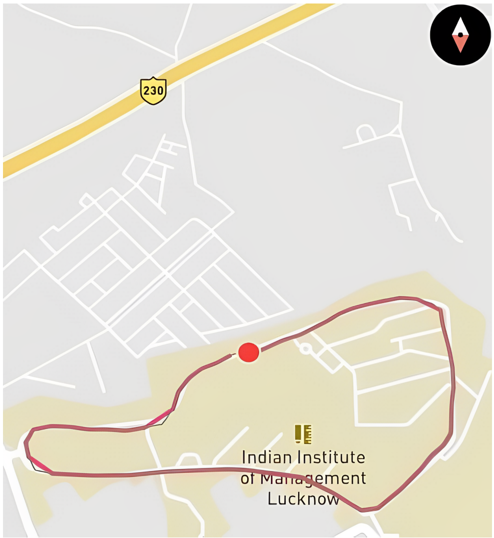

Fig. 11 Path Testing

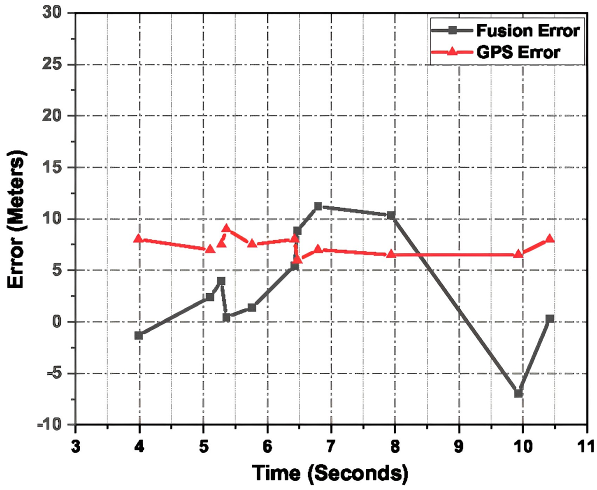

Fig. 12 Evaluation with .2HZ

rounds of data collection on the same route due to the changes in the traffic dynamics, driving conditions etc. are hardly reported.

7.1.3 Scenario 3: GPS frequency - 0.033 Hz

The object testing route and error values with corresponding timestamps at 0.033 HZ GPS frequency, are reported in the figure 13 and figure 14 respectively. As shown in the results, the proposed scheme has a slight edge over the GPS tracking method over the course of the experiment.

A careful observation of the reported results, show that the error values for the proposed scheme are higher at 0.033 Hz as compared to the other higher frequency values. A simple explanation for the reported error is the significantly low operating frequency.The proposed scheme, however, is able to outperform the GPS based method in most of the cases and achieves a reliable performance.

Reviewer 2: Comment 3

Fig. 13 Path Testing

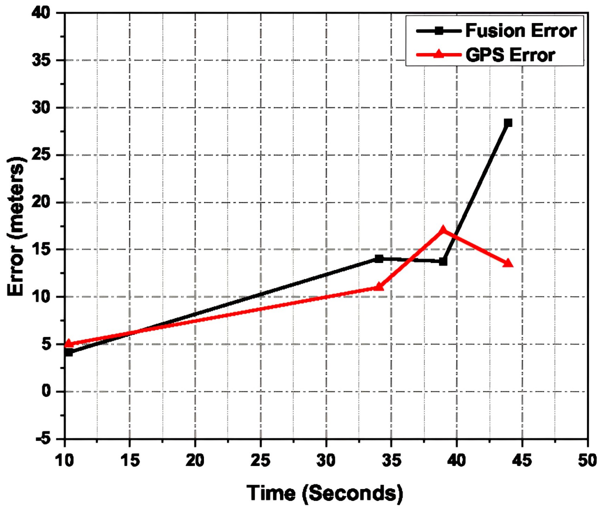

Fig. 14 Evaluation with .033HZ

7.2 Comparative Analysis:

A comprehensive comparative analysis of the proposed scheme is presented on two parameters, namely: Error Percentage and Root Mean Square Error (RMSE). The results of the proposed approach are compared with the scheme proposed in [10] and [11] for Error Percentage and RMSE, respectively.

Reviewer 2: Comment 7

7.2.1 Error Percentage

The table 3 clearly shows that the proposed scheme is able to outperform the approach proposed in [10] by a significant margin at almost all the GPS frequencies. As evident from the table 3 and figure 15, except at extremely low frequency of 0.033 Hz, the proposed scheme achieves an improvement of about 27% at 0.4 Hz. However, at extremely low frequency of 0.033 Hz, the reported error percentage of the proposed scheme is higher than that of the scheme proposed in [10] because the proposed scheme achieves the vehicle orientation and filtered vehicle speed, using the rotation vector based method.

Table 3 Comparison Between Proposed Method and [10] At Different Sampling Frequency

Frequency	Proposed Method %	Algorithm in[10]	Change %
.4HZ	40.56	55.09	27.04
.2HZ	44.43	69.03	35.64
0.033HZ	377.39	384.50	1.85

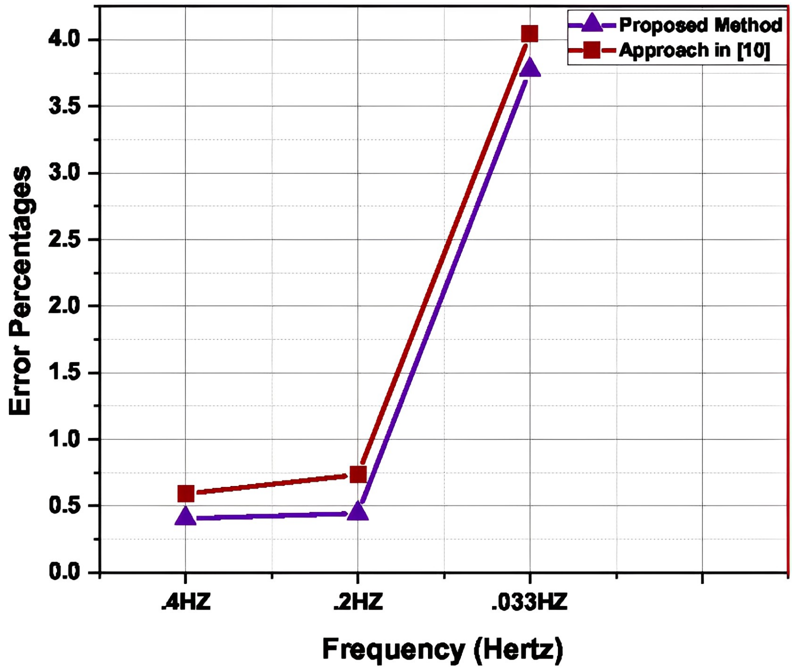

Fig. 15 Comparison Between Proposed Algorithm and [10] Based on Error Percentages

Reviewer 2: Comment 7

7.2.2 RMSE

The RMSE is calculated using the following formula:

$$RMSE = \sqrt{\frac{1}{N}[\sum_{n=j}^{N}(x_j - x_a)^2 + (y_j - y_a)^2} \tag{30}$$

The Table 4 and figure 16 shows a comparison of RMSE performance between proposed algorithm and [11], using two parameters: GPS Error and algorithm based error, for two paths (track 1 and track 2). The figure 12 reveals that proposed method achieves higher accuracy as compared to the approach in [11] because the algorithm in [11] provides vehicle speed and orientation using quaternion-based method but in case of the proposed method, rotation vector is used for orientation and double filtration for position estimation.

7.3 Run-time

The table 5 presents the run-time analysis of the proposed approach. As per the reported observations, approximately, 0.9 seconds is required by the proposed approach to complete the execution while the map box takes approximately 0.10 seconds,

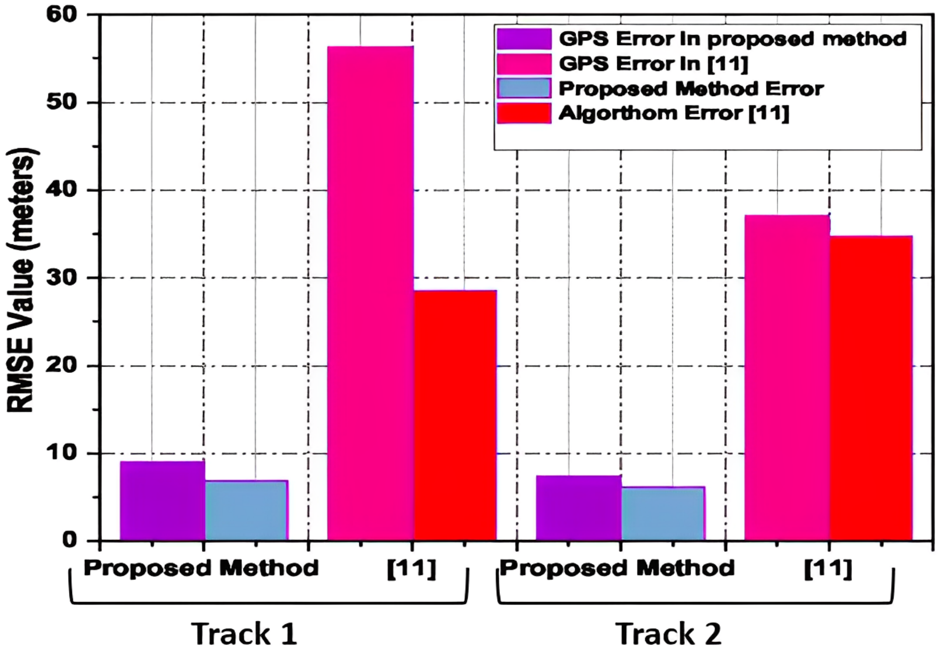

Fig. 16 Comparison Between Proposed Method and [11] Based on RMSE Performance.

Table 4 The RMSE Performance(meter) Between the Proposed Method and [11]

Track	Algorithm in [11]		Proposed Method	
	GPS	[11] Method	GPS	Proposed Method
1	56.34	28.86	9.017	6.82
2	37.06	34.73	7.41	6.14

thereby, making the overall time of the proposed approach to approximately 1 seconds to show the final output on the map. The data shows that the overall performance of the proposed method is acceptable in case of higher as well as for lower frequency sampling of GPS for object tracking.

8 Conclusion

A real-time application for object tracking based on multi sensor fusion is presented in this work. Combination of Kalman filter with Geo hash is used for determining the speed and position of the vehicle, while rotation vector used for orientation parametrisation which provides greater level of accuracy in orientation estimation. Extensive analysis of the proposed method,in different scenarios, proves the efficiency of the

Table 5 Time Response and Measurement Cycle

Task	Multi-Sensor fusion
Measurement Cycles	22 cycles per km
Average time	.5 sec

approach.The proposed method is able to outperform the existing approaches by a significant margin and achieves better accuracy in almost all the test cases.

Reviewer 4: Comment 3

9 Future Work

The obtained results show promise, although there are areas of improvement for future work. One aspect that could be enhanced is the geohash filtering technique, which primarily serves spatial indexing and efficient retrieval of geographic data based on proximity. To further enhance the system, a more advanced map matching technique could be introduced. This could involve considering previous locations or implementing a hidden Markov chain approach, which has the potential to yield improved outcomes.

Data Availability Statement

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Conflict of Interest (COI) Statement

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] P. Yang, D. Duan, C. Chen, X. Cheng, and L. Yang, "Multi-Sensor Multi-Vehicle (MSMV) Localization and Mobility Tracking for Autonomous Driving," IEEE Trans. Veh. Technol., pp. 1–1, 2020, doi: 10.1109/TVT.2020.3031900.

[2] G. Manogaran et al., "Multi-Variate Data Fusion Technique for Reducing Sensor Errors in Intelligent Transportation Systems," IEEE Sens. J., pp. 1–1, 2020, doi: 10.1109/JSEN.2020.3017384.

[3] A. Ali, Y. Zhu, and M. Zakarya, "A data aggregation based approach to exploit dynamic spatio-temporal correlations for citywide crowd flows prediction in fog

computing," Multimed Tools Appl, vol. 80, no. 20, pp. 31401–31433, Aug. 2021, doi: 10.1007/s11042-020-10486-4.

[4] A. Ali, Y. Zhu, and M. Zakarya, "Exploiting dynamic spatio-temporal correlations for citywide traffic flow prediction using attention based neural networks," Information Sciences, vol. 577, pp. 852–870, Oct. 2021, doi: 10.1016/j.ins.2021.08.042.

[5] M. Kok, J. D. Hol, and T. B. Schön, "Using Inertial Sensors for Position and Orientation Estimation," Found. Trends® Signal Process., vol. 11, no. 1–2, pp. 1–153, 2017, doi: 10.1561/2000000094.

[6] J. Permatasari, T. Connie, and T. S. Ong, "Inertial sensor fusion for gait recognition with symmetric positive definite Gaussian kernels analysis," Multimed Tools Appl, vol. 79, no. 43–44, pp. 32665–32692, Nov. 2020, doi: 10.1007/s11042-020-09438-9.

[7] C. Li, L. Yu, and S. Fei, "Real-Time 3D Motion Tracking and Reconstruction System Using Camera and IMU Sensors," IEEE Sens. J., vol. 19, no. 15, pp. 6460–6466, Aug. 2019, doi: 10.1109/JSEN.2019.2907716.

[8] R. V. Vitali, R. S. McGinnis, and N. C. Perkins, "Robust Error-State Kalman Filter for Estimating IMU Orientation," IEEE Sens. J., pp. 1–1, 2020, doi: 10.1109/JSEN.2020.3026895.

[9] R. Gobi, "Smartphone based indoor localization and tracking model using bat algorithm and Kalman filter," Multimed Tools Appl, vol. 80, no. 10, pp. 15377–15390, Apr. 2021, doi: 10.1007/s11042-020-10438-y.

[10] S. Plangi, A. Hadachi, A. Lind, and A. Bensrhair, "Real-Time Vehicles Tracking Based on Mobile Multi-Sensor Fusion," IEEE Sens. J., vol. 18, no. 24, pp. 10077–10084, Dec. 2018, doi: 10.1109/JSEN.2018.2873050.

[11] E. Choi and S. Chang, "A consumer tracking estimator for vehicles in GPS-free environments," IEEE Trans. Consumer Electron., vol. 63, no. 4, pp. 450–458, Nov. 2017, doi: 10.1109/TCE.2017.015064.

[12] A. Ibisch et al., "Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded LIDAR sensors," in 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast City, Australia, Jun. 2013, pp. 829–834, doi: 10.1109/IVS.2013.6629569.

[13] P. B. Fleischer, A. Y. Nelson, R.A. Sowah, and A. Bremang, "Design and development of GPS/GSM based vehicle tracking and alert system for commercial inter-city buses," in 2012 IEEE 4th International Conference on Adaptive Science Technology (ICAST), 2012, pp. 1–6.

[14] D. C. Obediencia, J. Kristoffer Pedrero, C. Villablanca, and M. A. Reyes, “eMap: Mobile App: Map Directory of EVSU Main Campus with Web-Based Management Panel,” in 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Laoag, Philippines, Nov. 2019, pp. 1–7, doi: 10.1109/HNICEM48295.2019.9073452.

[15] A. Celesti, A. Galletta, L. Carnevale, M. Fazio, A. Ĺay-Ekuakille, and M. Villari, “An IoT cloud system for traffic monitoring and vehicular accidents prevention based on mobile sensor data processing,” IEEE Sens. J., vol. 18, no. 12, pp. 4795–4802, 2017.

[16] C.-Y. Lin and F.-L. Lian, “System Integration of Sensor-Fusion Localization Tasks Using Vision-Based Driving Lane Detection and Road-Marker Recognition,” IEEE Systems Journal, vol. 14, no. 3, pp. 4523–4534, Sep. 2020, doi: 10.1109/JSYST.2019.2960193.

[17] R. Faragher, “Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation [Lecture Notes],” IEEE Signal Process. Mag., vol. 29, no. 5, pp. 128–132, Sep. 2012, doi: 10.1109/MSP.2012.2203621

[18] N. Guo, W. Xiong, Y. Wu, L. Chen, and N. Jing, “A Geographic Meshing and Coding Method Based on Adaptive Hilbert-Geohash,” IEEE Access, vol. 7, pp. 39815–39825, 2019, doi: 10.1109/ACCESS.2019.2906871.