Biomimetic Intelligence and Robotics 4 (2024) 100140

Contents lists available at ScienceDirect

Biomimetic Intelligence and Robotics

journal homepage: www.elsevier.com/locate/birob

Research Article

LiDAR-based estimation of bounding box coordinates using Gaussian process regression and particle swarm optimization

Vinodha K. *, E.S. Gopi, Tushar Agnibhoj

Department of Electronics and Communication Engineering, National Institute of Technology Tiruchirappalli, Tamil Nadu, 620015, India

ARTICLE INFO

Article history:
Received 4 August 2023
Revised 20 November 2023
Accepted 23 November 2023
Available online 27 November 2023

Keywords:
LiDAR
Data acquisition
Bounding box
Gaussian process regression
Particle swarm optimization (PSO)

ABSTRACT

Camera-based object tracking systems in a given closed environment lack privacy and confidentiality. In this study, light detection and ranging (LiDAR) was applied to track objects similar to the camera tracking in a closed environment, guaranteeing privacy and confidentiality. The primary objective was to demonstrate the efficacy of the proposed technique through carefully designed experiments conducted using two scenarios. In Scenario I, the study illustrates the capability of the proposed technique to detect the locations of multiple objects positioned on a flat surface, achieved by analyzing LiDAR data collected from several locations within the closed environment. Scenario II demonstrates the effectiveness of the proposed technique in detecting multiple objects using LiDAR data obtained from a single, fixed location. Real-time experiments are conducted with human subjects navigating predefined paths. Three individuals move within an environment, while LiDAR, fixed at the center, dynamically tracks and identifies their locations at multiple instances. Results demonstrate that a single, strategically positioned LiDAR can adeptly detect objects in motion around it. Furthermore, this study provides a comparison of various regression techniques for predicting bounding box coordinates. Gaussian process regression (GPR), combined with particle swarm optimization (PSO) for prediction, achieves the lowest prediction mean square error of all the regression techniques examined at 0.01. Hyperparameter tuning of GPR using PSO significantly minimizes the regression error. Results of the experiment pave the way for its extension to various real-time applications such as crowd management in malls, surveillance systems, and various Internet of Things scenarios.

© 2023 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

In the computer vision application landscapes, the foundational task of object detection has long relied upon cameras as the primary sensing modality [1,2]. However, persistent challenges, such as sensitivity to lighting variations and privacy concerns, have fueled the exploration of alternative methodologies. This study supports a transformative shift from the traditional use of cameras by proposing the integration of two-dimensional (2D) light detection and ranging (LiDAR) sensors for object detection, a shift underscored by notable advancements in sensor technology and the proliferation of deep learning methodologies [3–7].

1.1. Research background

Research builds upon an evolving environment, highlighting the significance of LiDAR usage and the fusion of deep learning techniques in object and human detection across diverse environments. LiDAR-based inspection, exemplified by its efficacy in identifying flaws in bridge construction, enhances operational efficiency and mitigates the risks associated with the presence of humans in high-risk locations [8]. Techniques involving beam-steering and scene flashing provide valuable insights into optical modulation, LiDAR designs, and detection methods, broadening the LiDAR application spectrum [9].

Regarding mobile robot navigation, prior attempts utilizing support vector machine classification demonstrated commendable accuracy in indoor object detection [3,10]. However, challenges persist in the precise location of objects within complex scenes, fueling the need for advancements in segmentation methods [11]. Despite strides in these methodologies, acquiring pinpoint accuracy in object location remains an ongoing challenge. Comparative studies of 2D and 3D LiDAR data for pedestrian detection demonstrate the comparable or superior results obtained with the 2D LiDAR data [12,13]. Combining 2D LiDAR with camera data to estimate the indoor environment layouts, leveraging motion cues, and optimization algorithms demonstrates the versatility of this sensing modality [14]. Moreover, machine-learning algorithms applied to 2D LiDAR data analysis proved effective in fall detection and activities of daily living monitoring, emphasizing the adaptability of this technology [15].

* Corresponding author.
E-mail address: vinodhakamaraj@gmail.com (Vinodha K.).

https://doi.org/10.1016/j.birob.2023.100140
2667-3797/© 2023 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1.2. Research significance

Despite these advancements, challenges persist in the existing LiDAR data processing methods related to computational complexity and privacy. In this study, this gap is addressed by introducing an innovative approach centered on estimating object coordinates. This approach involves using camera target values derived from LiDAR data while focusing on error minimization and reducing reliance on visible light. The potential impact of this research extends beyond computer vision, with the potential to transform surveillance, crowd management, and various IoT applications. The proposed methodology could disrupt the surveillance landscape, providing a compelling alternative to traditional CCTV cameras. Meanwhile, it ensures privacy in object detection applications and enhances accuracy under challenging lighting conditions, positioning this research as a motivator in reshaping surveillance technologies.

The indoor LiDAR-based object detection determines the target points by tracking specific features or objects located in the LiDAR point cloud data. One unexplored application is the estimation of object coordinates using the camera-derived target values, suggesting a new research direction and facilitating the replacement of CCTV cameras with LiDAR for surveillance purposes. An experimental dataset including LiDAR data and camera-derived coordinates is provided for public use. During training, the system uses camera data to establish target points; however, during testing, the system operates independently of cameras, relying solely on LiDAR data. This methodology is unique and significant in advancing LiDAR-based surveillance technologies.

1.3. Overview of the proposed work

To our knowledge, no attempts have been made to estimate object positions using the 2D LiDAR by estimating camera target values for determining object coordinates. This study demonstrated that Gaussian process regression (GPR) with hyperparameters tuned by particle swarm optimization (PSO) gives better results than state-of-the-art techniques, such as kernel smoothing regression and convolution neural networks (CNN), in lidar applications. In GPR, hyperparameters can be tuned using various methods, including manual tuning, grid search, random search, Bayesian optimization, maximum likelihood estimation [16], and PSO. In this study, PSO is adopted to tune hyperparameters in LiDAR-based applications and minimize object detection errors in closed environments. In this work, demonstrating 2D LiDAR, data are collected with 360° coverage and the object position at various locations is identified using regression techniques with an optimization algorithm. Data preprocessing is conducted, applying interpolation of the LiDAR data and morphological operations on the binary image data.

The main contributions of this study are as follows:

(1) The usage of 2D LiDAR data to estimate the coordinates of the bounding boxes of multiple objects obtained from a digital camera in closed environment scenarios is demonstrated as a proof of concept.
(2) A methodology for predicting the bounding box coordinates using GPR techniques is established and compared with other regression techniques.
(3) An objective function applying PSO to tune the hyperparameters used in GPR is formulated.
(4) A benchmark LiDAR dataset based on experiments conducted under two scenarios (with varying size boxes and human subjects as target objects) is provided for follow-up research.

2. Proposed methodology

In a typical environment, LiDAR data are recorded using a sensor, and the corresponding image data (top view) are obtained using a camera. The acquired LiDAR data was preprocessed using an interpolation technique and arranged in the matrix form to obtain the input data. Similarly, raw top-view images were subjected to morphological operations, and bounding boxes were obtained using coordinate values. Top-left and bottom-right coordinates of the individual bounding boxes were collected as the corresponding target output data (Fig. 1). Post preprocessing stage, input and corresponding target data were obtained, and data normalization was performed. The experiment was then repeated to collect various input and corresponding output data. Ultimately, the GPR model was established using the collected data and compared with other regression model techniques, namely, Kernel smoothing and CNN.

2.1. LiDAR sensor description

LiDAR is a remote sensing technology that uses lasers to measure the distance of objects in a scene. LiDAR sensors emit laser pulses and measure the time required for the pulses to return after bouncing off objects in the environment. In this study, a YDLIDAR X4 sensor was used to collect object data at different locations. It rotated 360° to scan and range the surrounding environment, producing a map outline. The YDLIDAR X4 frequency range was 5000 Hz with 6–12 Hz scan frequency, and the ϕ size was 65.6 × 58.39 × 101.7 mm. The resolution angle and distance range were 0.43–0.860 and 0.12–10 m, respectively. YDLIDAR provides a point cloud viewer (LidarViewer V0.1.9), a visualization program for X4 real-time scanning, shows in Fig. 2.

According to the product description for the YDLIDAR X4 sensor, the ideal working environment is indoors, and light in the indoor environment does not affect the performance of the device, confirmed by experiments conducted with the test data in real-time with various light intensities (LUX → luminous), such as 312, 161, and 26 LUX. Furthermore, it was demonstrated to detect objects effectively, even in a dark room. Thus, YDLIDAR X4 can function regardless of ambient light conditions; it does not rely on the presence of visible light to operate, making it suitable for a wide range of lighting conditions. To validate the proposed technique, the research experiment is performed under two different scenarios in an indoor environment.

2.2. Input data preparation obtained from LiDAR sensor

The LiDAR sensor begins to scan and display the environment point cloud data. The upper left corner in Fig. 2 shows the angle and distance information of the nearby object. In the processing of the LiDAR data, interpolation is performed to create a continuous representation of the data and the sample size is standardized. The angles are fixed at regular intervals, permitting the retention of the distance data for further processing. The original data size in Scenario I varies from 450 to 850 distance measurements and from 450 to 750 in Scenario II. After interpolation, the data are standardized to have equal angles at fixed intervals. The interpolated data size is a 14 × 900 matrix (14 LiDAR positions, 900 distance measurements for each position) in Scenario I and 1 × 800 matrix in Scenario II (LiDAR positioned at the center, with the data point at each instance equal to 800 after interpolation). In this Scenario, the data are split into four rows, each with 200 points.

The datasets are processed separately in scenarios I and II, resulting in different numbers of datasets and corresponding input matrix sizes. For Scenario I, the number of data entries is

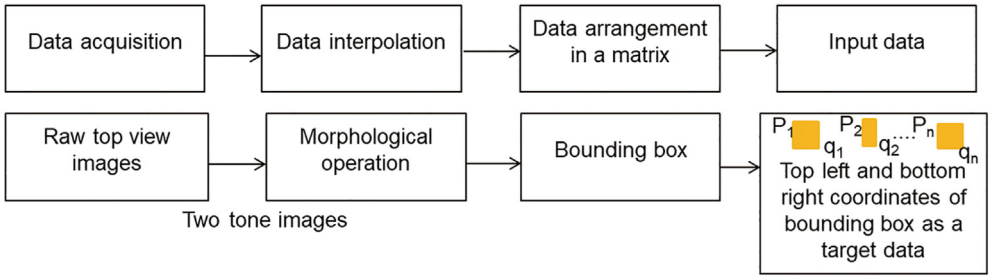

Fig. 1. Block diagram of the proposed technique.

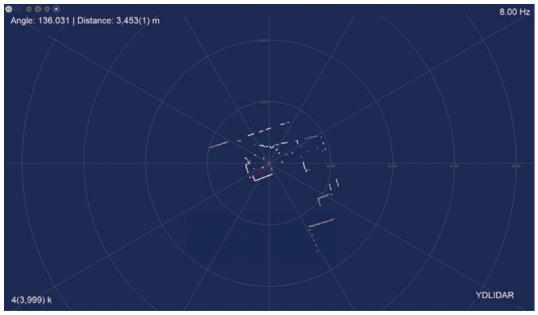

Fig. 2. Data visualization tool called LidarViewer for X4 real-time scanning of objects positioned at various locations.

700, while in case II, it is 100. These standardized and processed data matrices, each representing a single LiDAR dataset with distance measurements at fixed-angle intervals, can now be used for further analysis and modeling. Depending on the specific goals, they can be applied to various statistical or machine-learning techniques to gain insights or make predictions. The uniformity in the data size and interpolation step could facilitate the use of the data and enable consistent analysis across different datasets.

2.3. Target data preparation obtained from mobile camera

The coordinate points of the bounding box (Top-left and bottom-right corners) corresponding to the individual objects in the captured image data are obtained (as described in Fig. 3). The RGB image format is converted to HSV, and a matrix (with $0.12 < H < 0.18$) is obtained for further operation.

The combinations for Scenario II are as follows:

- ((r>155) & (r<210)) & ((g>135) & (g<190)) & ((b>60) & (b<120))
- ((r>190) & (r<230)) & ((g>150) & (g<210)) & ((b>95) & (b<150))
- ((r>180) & (r<230)) & ((g>180) & (g<230)) & ((b>85) & (b<160))

The morphological processes of erosion and dilation are performed on the image to remove the unwanted details of the object and, more precisely, retrieve the area of interest (target object). Morphological operations modify the shape or structure of objects in an image. The procedure involves enlarging or contracting the boundaries of the objects in an image to isolate particular features or eliminate noise or other undesirable elements. Dilation involves the expansion of the boundaries of objects in an image by adding pixels to the object edges. It can be useful for connecting broken objects or emphasizing an object's features. Erosion involves the shrinking of an object's boundaries in an image by removing pixels from the edges of the objects. It can help to eliminate small, isolated pixels or separate overlapping objects. Specifically, a mask of size 15×15 is used to perform erosion followed by dilation to obtain a filtered image with reduced noise (Fig. 4). A bounding box is a rectangular area that "bounds" or encloses an object in an image. In computer vision and machine-learning applications, bounding boxes are frequently used to locate and identify objects in an image.

In this study, a bounding box is created around a rectangular object in an image and used to identify the coordinates of the object's top-left and bottom-right corners. The minimum and maximum values of the x- and y-coordinates of these points define the bounding box in Scenario I. Similarly, in Scenario II, the top-left and bottom-right coordinates of an object define its bounding box. Once a bounding box has been created, it becomes easy to locate the object. A summary of the size of the dataset used for regression is tabulated in Table 1, where $* \rightarrow$ one for each regression model.

3. Construction of regression models

Regression techniques are applied to LiDAR data to identify the position of objects at various locations.

3.1. Gaussian process regression

GPR is a nonparametric regression technique that assumes the data as random variables follow a multivariate normal distribution.

Let $y_1, y_2, y_3 \cdots y_N$ be the noisy target observations corresponding to the input vectors $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3} \cdots \mathbf{x_N}$ for N instances. $y(\mathbf{x}) = \phi(\mathbf{x}) + \epsilon$ where, $\mathbf{x} \rightarrow$input vector, $y \rightarrow$corresponding noisy observation, $\epsilon \rightarrow$ Gaussian-distributed random variable with a mean zero and variance $\frac{1}{\lambda}$, and $\phi(\mathbf{x})$ is the nonlinear basis function. This training dataset is the typical outcome of the correlated discrete random process (Fig. 5). The predicted value corresponding to the test input vector ($\mathbf{x_{test}}$) is obtained as the conditional mean of the random variable y_{test}, conditioned on a typical outcome of the random process (training dataset). The joint density function of random vector $[y_1 \ y_2 \ y_3 \cdots y_N \ y_{test}]$ can be shown to follow a multivariate Gaussian distribution with mean zero and covariance matrix $\mathbf{C}$.

$$\mathbf{y} = [y_1 \ y_2 \ \ldots y_N]^T \tag{1}$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C_y} & \mathbf{k_{test}} \\ \mathbf{k_{test}^T} & k(\mathbf{x_{test}}, \mathbf{x_{test}}) + \frac{1}{\lambda} \end{bmatrix} \tag{2}$$

where

$$\mathbf{C_y} = \begin{bmatrix} k(\mathbf{x_1}, \mathbf{x_1}) + \frac{1}{\lambda} & \cdots & k(\mathbf{x_1}, \mathbf{x_N}) \\ k(\mathbf{x_2}, \mathbf{x_1}) & k(\mathbf{x_2}, \mathbf{x_2}) + \frac{1}{\lambda} & \vdots \\ \vdots & & \vdots \\ k(\mathbf{x_N}, \mathbf{x_1}) & & k(\mathbf{x_N}, \mathbf{x_N}) + \frac{1}{\lambda} \end{bmatrix} \tag{3}$$

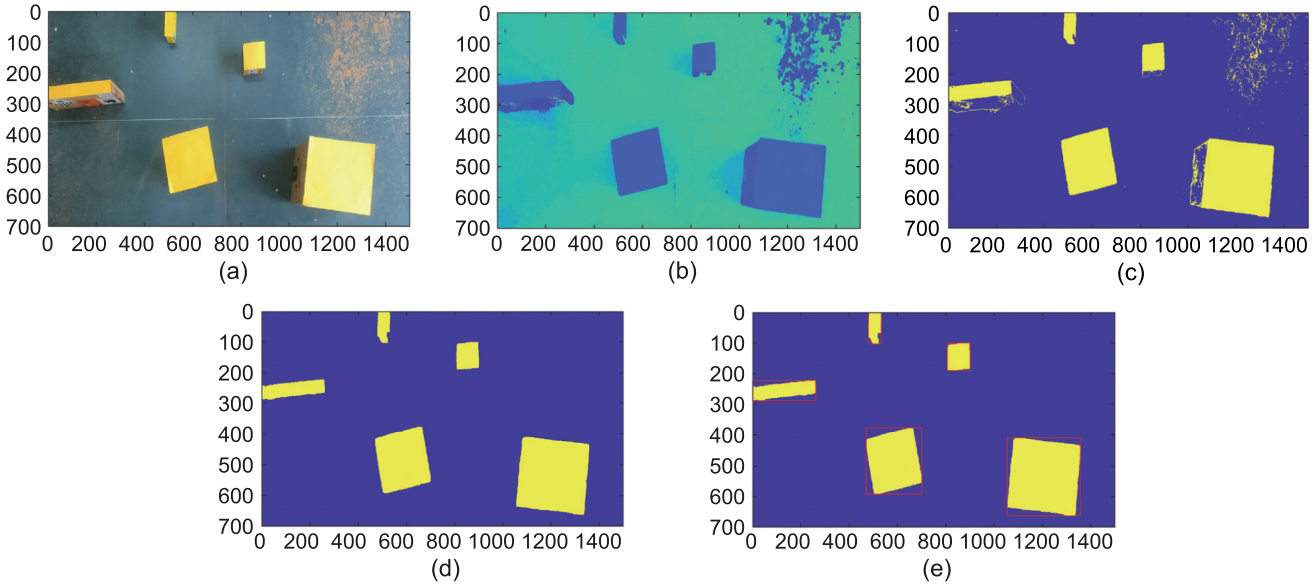

Fig. 3. Illustrations of the image preprocessing steps to obtain the coordinates of the target objects (target coordinates for regression model) for Scenario I. (a) Top view of raw image. (b) HSV image. (c) Two-tone top-view image. (d) After erosion and dilation operation. (e) Bounding box on the image used to acquire the diagonal points.

Table 1
Summarization of the dataset size (input and corresponding target) for various regression techniques.

Scenarios	GPR		CNN		KSR	
	Input	Target	Input	Target	Input	Target
Scenario I (20 Regression models)	25 × 25*	1 × 1	14 × 900	20 × 1	1 × 12,600*	20 × 1
Scenario II (12 Regression models)	50 × 50*	1 × 1	4 × 200	12 × 1	1 × 800*	12 × 1

$$\mathbf{k_{test}} = \begin{bmatrix} k(\mathbf{x_1}, \mathbf{x_{test}}) \\ \vdots \\ k(\mathbf{x_N}, \mathbf{x_{test}}) \end{bmatrix} \tag{4}$$

and $k(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i}^T)\phi(\mathbf{x_j})$. It can be demonstrated that the conditional density function, denoted by the notation $f(y_{test}/\mathbf{x_1}\ \mathbf{x_2}\ \mathbf{x_3} \ldots \mathbf{x_N}\ y_1\ y_2\ y_3 \ldots y_N\ \mathbf{x_{test}})$, has a Gaussian distribution with a mean of

$$\hat{y}_j^{test} = \mathbf{k_{test}^T}(\mathbf{C_y})^{-1}\mathbf{y} \tag{5}$$

and variance $k(\mathbf{x_{test}}, \mathbf{x_{test}}) + \frac{1}{\lambda} - \mathbf{k_{test}^T}(\mathbf{C_y})^{-1}\mathbf{k_{test}}$. Furthermore, $\mathbf{C}_y = \mathbf{K} + \frac{1}{\lambda}\mathbf{I}$, where $K = \phi^T\phi$ denotes the kernel smoothing matrix, and $\mathbf{k}(\mathbf{x_{test}}, \mathbf{x_i})$ denotes the kernel function $\phi(\mathbf{x_{test}})^T\phi(\mathbf{x_i})$. $k(\mathbf{x_m}, \mathbf{x_n})$ represents the (m, n)th element of the kernel smoothing function. Below is presented the generated kernel function.

$$k(\mathbf{x_i}, \mathbf{x_k}) = \theta_0\, e^{-\frac{\theta_1}{2}\|(\mathbf{x_i}-\mathbf{x_k})^2\| + \theta_2 + \theta_3(\mathbf{x_i^T x_k})} \tag{6}$$

where the hyperparameters are $\theta_0, \theta_1, \theta_2, \theta_3$, optimized using PSO. The steps in GPR in our proposed work are shown in Algorithm 1.

PSO is a heuristic optimization algorithm used to find the global minimum of a given objective function [17]. The algorithm iteratively updates a population of particles in a search space, in which each particle represents a potential solution to the problem [17,18]. The Algorithm 2 combines individual and collective learning to move the particles toward the global minimum.

The probability that the individual test data $\mathbf{y_j^{test}}$, generated from the multivariate Gaussian density function model, depends upon the covariance matrix ($\mathbf{c_y}$) and mean vector $\hat{\mathbf{y}}_\mathbf{j}^{\mathbf{test}}$. These are constructed using a kernel function that depends upon the hyperparameters $\theta_0, \theta_1, \theta_2, \theta_3$. The function J_1 (Eq. (7)) is formulated as the product of the probability density functional values of all the individual test data. Thus, the minimization problem is constructed using the fitness function $J = \frac{1}{J_1}$ as follows:

$$J_1 = \prod_{j=1}^{M} \frac{1}{(2\pi)^{M/2}|\mathbf{c_y}|} e^{-\frac{1}{2}[\mathbf{y_j^{test}} - \hat{\mathbf{y}}_\mathbf{j}^{\mathbf{test}}]^T \mathbf{c_y}^{-1} [\mathbf{y_j^{test}} - \hat{\mathbf{y}}_\mathbf{j}^{\mathbf{test}}]} \tag{7}$$

$$J = \underset{\theta_0\,\theta_1\,\theta_2\,\theta_3}{\arg\ min} \frac{1}{J_1} \tag{8}$$

Algorithm 1 Gaussian process regression

Require:
$\mathbf{x_1}\ \mathbf{x_2} \ldots \mathbf{x_N}$ ▷ Train data
$y_1\ y_2 \ldots y_N$
$\mathbf{x_1^{test}}\ \mathbf{x_2^{test}} \ldots \mathbf{x_M^{test}}$ ▷ Test data
$y_1^{test}\ y_2^{test} \ldots y_M^{test}$
Ensure: $\theta_0,\ \theta_1,\ \theta_2,\ \theta_3,\ \lambda$ ▷ Find $\theta_0, \theta_1, \theta_2, \theta_3$ such that J is minimized
function $(k(\mathbf{x_i}, \mathbf{x_k}) = \theta_0\, e^{-\frac{\theta_1}{2}\|(\mathbf{x_i}-\mathbf{x_k})^2\| + \theta_2 + \theta_3(\mathbf{x_i^T x_k})})$
Compute the covariance matrix C_y using the training data for the particular $\theta_0, \theta_1, \theta_2, \theta_3$ (Eq. (3))
for $j \leftarrow 1$ to M **do**
Compute k_{test} corresponding to $\mathbf{x_j^{test}}$ (Eq. (4))
$\hat{y}_j^{test} \leftarrow \mathbf{k_{test}^T}(\mathbf{C_y})^{-1}\mathbf{y}$ ▷ Compute the predicted value $\hat{y}_j^{test}$
end for
end function

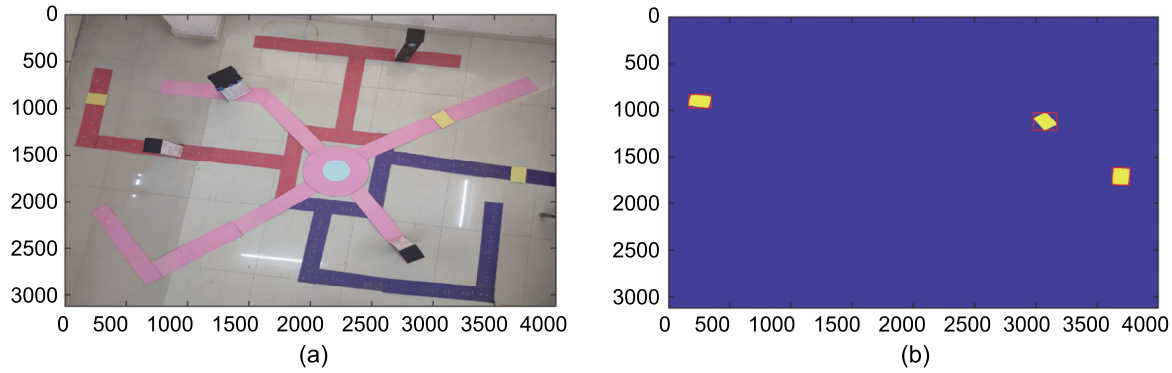

Fig. 4. Illustrations of the preparation of the target data for Scenario II. The three different color strips (red, pink, and violet) indicate the tracking paths of three objects. The black-covered box represents the obstacle whose location is constant in all instances, while the yellow patches show the positions of the object (human), whose location changes at each instant. The top view of the (a) raw image and (b) region of interest (bounding box on the image to collect the top-left and bottom-right coordinates of the bounding box).

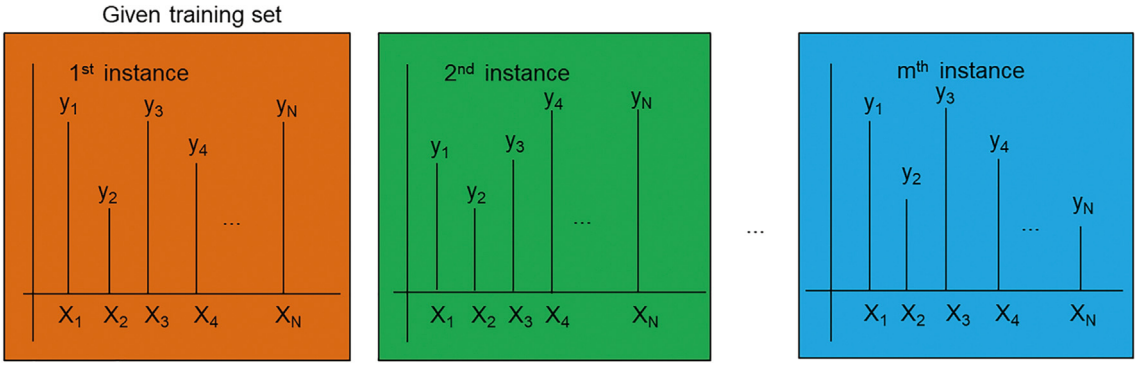

Fig. 5. Illustration of the observed training set (1st instance) as one of the outcomes of a discrete correlated random process with a mean vector of $[\phi(\mathbf{x_1})\cdots\phi(\mathbf{x_N})]^{-1}$ and correlation matrix of $\mathbf{C_y}$.

Algorithm 2 Particle swarm optimization with fitness function

```
Load input and target data
Load the best values and center point
Initialize the particle population
Set the maximum number of iterations and convergence threshold
Initialize the global and local best positions and fitness values
while the number of iterations is less than the maximum allowed and the convergence criteria are not met do
    for each particle in the population do
        Update the velocity and position of the particle
        Evaluate the fitness function using the new position
        if the fitness value is better than the local best then
            Update the local best position and fitness value
        end if
        if the fitness value is better than the global best then
            Update the global best position and fitness value
        end if
    end for
    if the difference between the previous and current global best fitness values is less than the convergence threshold then
        Terminate the algorithm
    end if
end while
Return the global best position and fitness value
```

3.2. Regression model using kernel smoothing method

The Algorithm 3 shows the procedure for the kernel smoothing routine used in this study. LiDAR data coordinate estimation of objects is obtained by applying the kernel smoothing technique. The $1 \times m$ input data corresponding to t_n target points are obtained from the diagonal coordinates of the bounding box image. The dataset is divided into 50% for training and the remaining for testing. Of the testing data, 25% are for validation and 25% reserved for testing. Initially, the Gaussian kernel function with the training and testing data within a specified range of sigma from 0.01 to 50 is used to retrieve each target point. Each target point is obtained by regressively adding the target to the result of the kernel function and the target specified for the training data. According to the kernel smoothing algorithm, each point is calculated from the validation data, with the σ value set to 0.8. Finally, the predicted target points and average error are determined using the test data.

$$\sum_{k=1}^{k=N_{test}} \left(\mathbf{t}_{\mathbf{test}}^{\mathbf{k}} - \sum_{i=1}^{N} K(\mathbf{x_i}, \mathbf{x_k})\mathbf{t_i}\right)^{T} \left(\mathbf{t}_{\mathbf{test}}^{\mathbf{k}} - \sum_{i=1}^{N} K(\mathbf{x_i}, \mathbf{x_k})\mathbf{t_i}\right) \tag{9}$$

where $\mathbf{x_i}$ stands for the training set, $\mathbf{t_i}$ for the matching target, $\mathbf{x_k}$ for the test vector, $\mathbf{t}_{\mathbf{test}}^{\mathbf{k}}$ for the estimated target, and N_{test} for the number of test data.

Algorithm 3 Kernel smoothing

Require:
$\mathbf{x_1}, \mathbf{x_2} \ldots \mathbf{x_N}$ ▷ training data
$\mathbf{t_1}, \mathbf{t_2} \ldots \mathbf{t_N}$
$\mathbf{x_1^{test}}, \mathbf{x_2^{test}} \ldots \mathbf{x_M^{test}}$ ▷ test data
$\mathbf{t_1^{test}}, \mathbf{t_2^{test}} \ldots \mathbf{t_M^{test}}$
Ensure: σ
function $(\mathrm{k}(\mathbf{x_j}, \mathbf{x_{test}}, \sigma))$
$\mathrm{K}(\mathbf{x_j}, \mathbf{x_{test}}) \leftarrow \mathrm{e}^{\frac{-(\mathbf{x_j}-\mathbf{x_{test}})^{\mathrm{T}}(\mathbf{x_j}-\mathbf{x_{test}})}{\sigma^2}}$ ▷ Gaussian kernel function
$e \leftarrow 0$
for $j \leftarrow 1$ to M **do**
$s \leftarrow 0$
for $i \leftarrow 1$ to N **do**
$s \leftarrow \mathrm{s} + \mathrm{K}(\mathbf{x_i^{train}}, \mathbf{x_j^{test}}, \sigma) \times \mathbf{t_i^{train}}$
end for
$\mathbf{e} \leftarrow \mathbf{e} + (\mathbf{s} - \mathbf{t_j^{test}})^2$ ▷ e ← Error
end for
Find σ such that e is minimized
end function

3.3. Regression model using convolution neural networks

A deep learning model named CNN regression 1 is employed to forecast continuous-valued outputs, also known as "regressions". CNN regression models use convolutional filters through many layers to extract features from the input data. However, a CNN regression model generates a continuous-valued output representing a prediction or estimate of the desired quantity, rather than a class label or probability, as the final result. The network architecture is created with input data of n × m × 1 (n is the number of object locations) and a target of $\mathbf{t_n}$(number of x- and y-coordinate points of all the objects placed at one instant) points. The number of convolutional layers used to learn the input data is three. Convolutional filters are applied to the input data during the processing stage, with each filter activating different features from the data sample [19]. A "Relu layer" is used as an activation function, providing quicker and more effective training with positive values kept and negative values converted to zero. Then, pooling is performed to reduce the number of parameters needed by the network to learn through nonlinear downsampling of the output. A sigmoid layer is finally used at the end of the network before the regression layer to convert into a positive values.

K-fold cross-validation is used to validate the CNN model. The number of folds K is set to 5. K − 1 folds are chosen as the training dataset, and the rest as the test dataset. From "m" data samples, 50% are used to train the model as training data and the remaining are used to validate the model as test data. On each cross-validation iteration, we train a new model independent of the model trained on the previous iteration and then validate it with the test data, repeating the procedure K times. Each time, we use the remaining fold as the test set. The final score is calculated by taking the average of the iterations.

4. Experiments and results

Here, an approach to using LiDAR technology to track the object position within an indoor closed environment was proposed. The primary focus was to ensure privacy and confidentiality while achieving accurate object localization. Our study encompassed two scenarios to assess the efficacy of this technique, both of which yielded good results.

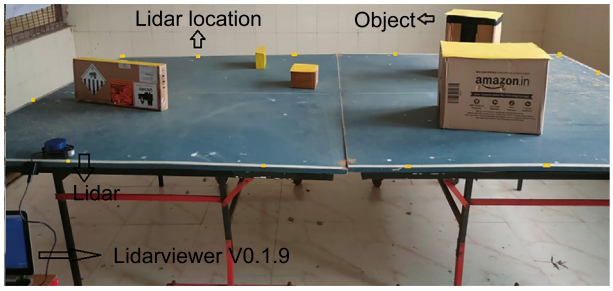

Fig. 6. Experimental setup of LiDAR data collection for Scenario I, the data collected from 14 multiple locations and with various objects.

4.1. Scenario I: LiDAR at multiple locations with multiple objects in a closed environment

Our proposed technique demonstrated the ability to detect and locate multiple objects on a flat surface with minimum detection error. By utilizing 2D-LiDAR data collected from various positions, the system effectively predicted the bounding box coordinates of these objects.

A table tennis table is used as a flat surface (prototype environment) (refer to Fig. 6), and cardboard boxes of different sizes are used as objects. Five objects of different sizes are placed at various locations on the table. A YDLIDAR X4 sensor collects the data at a different location. The LiDAR collects 360-degree data along with the angle, distance, and intensity, and 14 location points for the LiDAR are set up around the table.

The data from each location are collected using LiDAR viewer software. 50 sets at 14 different locations are compiled, accounting for 700 data samples acquired. The 700 data samples involve different numbers of observations; thus, interpolation is used to make the data size equal. Based on the angle, we arrange the distance such that each data file has the same sample size.

The compiled results show how close the predicted object positions are to the original positions using the regression technique of the study. Figs. 7, 8, and, 9 of the Scenario I dataset depicts two sample plots showing the locations of the original (acquired using the camera) and predicted target coordinates (using LiDAR).

The bounding boxes obtained using GPR for two randomly selected test data samples are illustrated in Fig. 10(a) and 10(b). The top-left coordinates, denoted as p and p', are obtained using camera and LiDAR, respectively; meanwhile, the bottom-right coordinates, denoted as q and q', are obtained using a camera and LiDAR, respectively. The blue rectangles represent the original bounding boxes, whereas the red rectangles denote the predicted bounding boxes. Notably, the proximity between the original and predicted bounding boxes shows a close alignment, indicating the accurate positioning of objects at various locations.

4.2. Scenario II: LiDAR at the center point (single location) with multiple objects in a closed environment

The performance of the technique in detecting multiple objects using LiDAR data from a single, fixed location was evaluated. The experiments involved the real-time tracking of human subjects within the LiDAR's observation area. The results showed that a strategically positioned single LiDAR device could successfully detect moving objects nearby.

An environmental path was set in a room where three people moved around a track. In Fig. 11, three different colored strips (red, pink, and violet) indicate the walking track that the humans use to move around the path. A black-covered object indicates an object whose position is the same in all instances. The lidar is

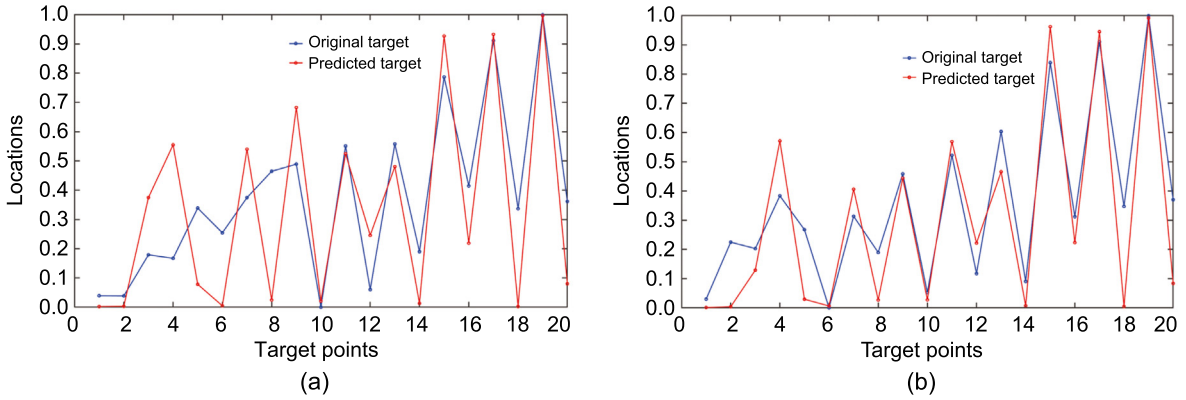

Fig. 7. Comparison of original (acquired using the camera) and CNN-based predicted coordinates using the LiDAR data for Scenario I. (a) Sample test data 1. (b) Sample test data 2.

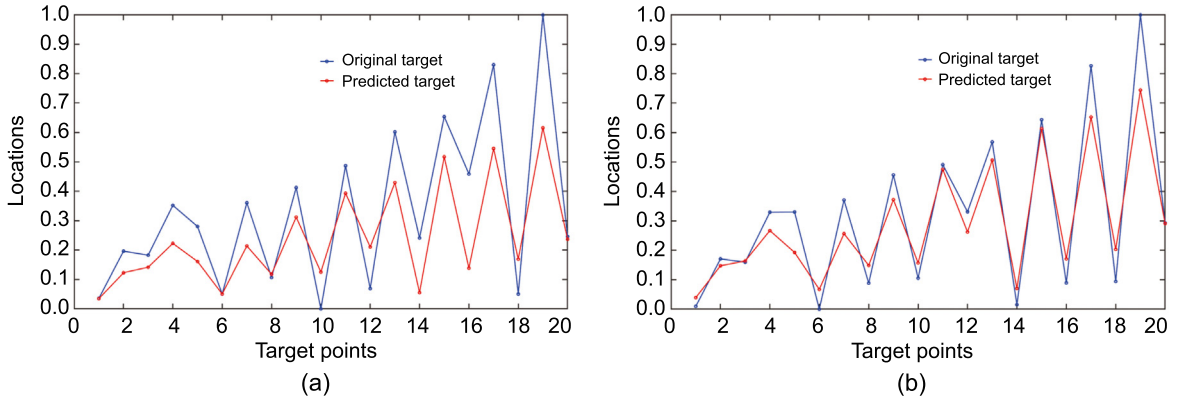

Fig. 8. Comparison of original (acquired using the camera) and Gaussian-smoothing-based predicted coordinates using LiDAR data for Scenario I. (a) Sample test data 1. (b) Sample test data 2.

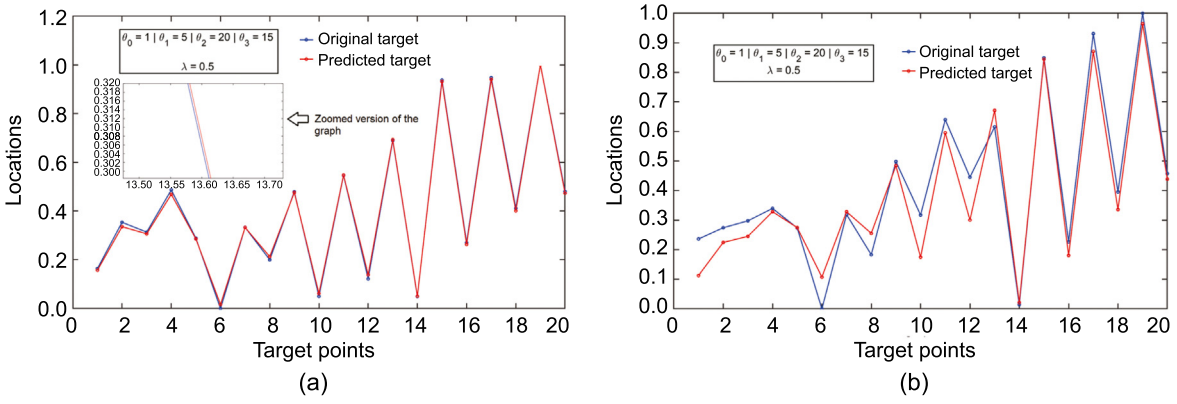

Fig. 9. Comparison of original (acquired using the camera) and predicted coordinates from GPR using LiDAR data for Scenario I. (a) Sample test data 1. (b) Sample test data 2.

fixed at the center, collecting 360° data. Data were collected for 14 instances (14 different times), and eight different positions of a moving object were collected at each instant.

Polynomial interpolation is used to obtain the length of the LiDAR data, identical to several attempts. The target points are collected for both environments in a closed space by mounting the Android mobile phone camera onto the top view. The camera captures the target image of multiple objects at different locations.

The data are distributed as training and testing sets in a ratio of 5:5, such that the training data includes 50 sets and the testing data comprise the remaining sets. A sample of the raw data and interpolated LiDAR data are shown in Fig. 12.

Furthermore, our research focused on the fine-tuning of the GPR hyperparameters using PSO, resulting in optimized regression error minimization. The experimental results were compared with those of various regression techniques, such as CNN, Kernel smoothing regression, and GPR with PSO to estimate the coordinates of an object at various locations.

The Gaussian process is computed on the dataset obtained from the LiDAR. Fig. 13 shows a representation of the covariance matrix $\mathbf{C_y}$ of the Gaussian process, calculated using Eq. (3). The

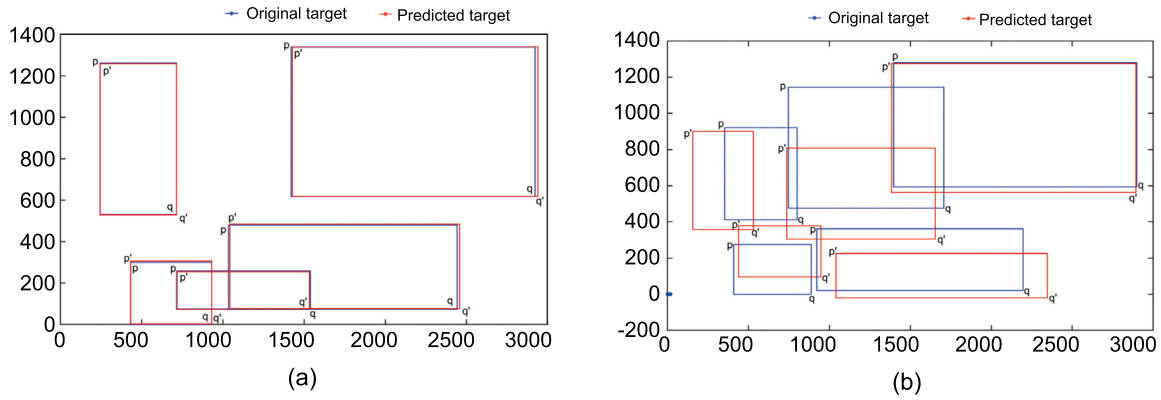

Fig. 10. Gaussian process regression was used to predict the top-left and bottom-right coordinates of the individual bounding boxes of five objects attended to by cameras (p and q, respectively), as well as p' and q', the top-left and bottom-right coordinates of the LiDAR objects. (a) Sample test data 1. (b) Sample test data 2 for Scenario I.

Fig. 11. Experimental setup of LiDAR data collection for Scenario II, whereby the LiDAR was placed in the middle of the closed environment with multiple objects. Three different color strips are used to indicate the path of the people moving about in various instances.

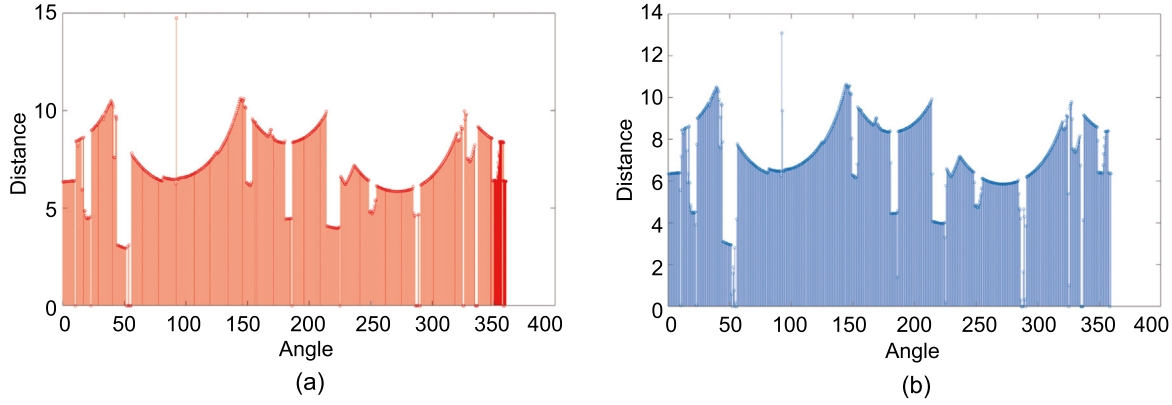

Fig. 12. Illustration of sample raw LiDAR test data and the polynomial interpolated LiDAR test data (distance versus angle) at one instance, which is used to estimate 12 coordinate values corresponding to three objects in Scenario II (refer to Eq. 12).

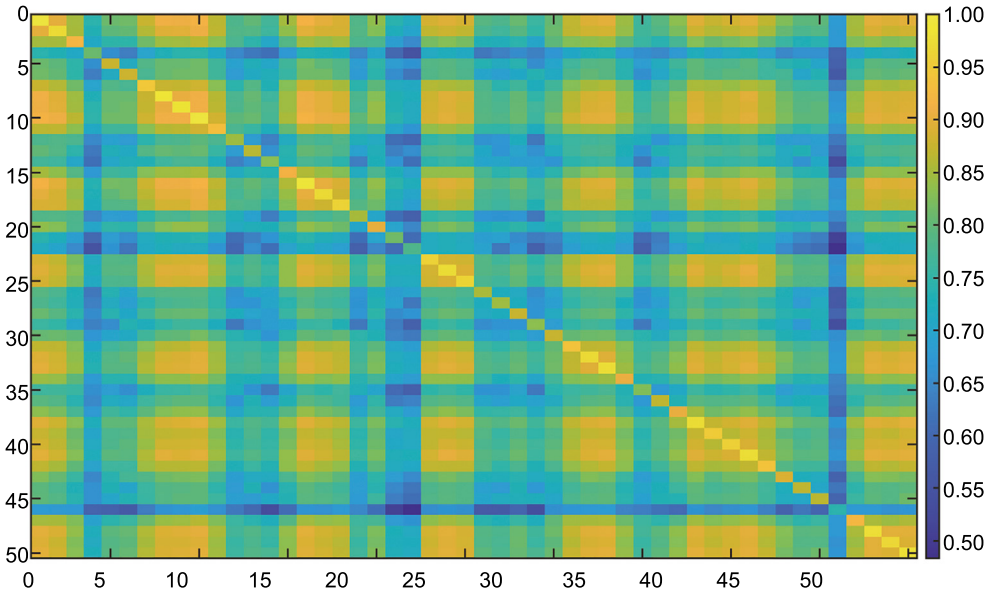

Fig. 13. Covariance matrix C_y of the Gaussian process obtained using 50 LiDAR training data sequences, which are used to estimate the coordinate value (refer to Eqs. (2), (3), (4) and (6)).

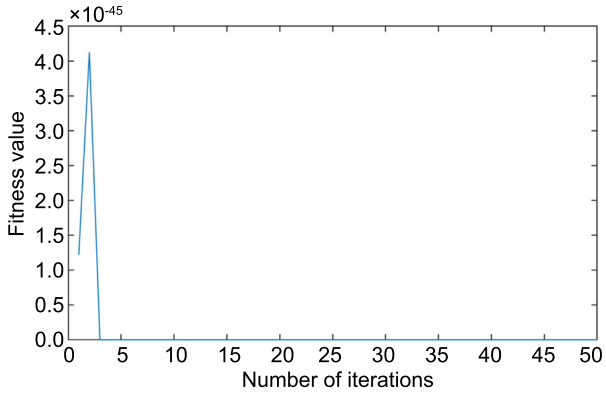

Fig. 14. Sample PSO convergence graph in Gaussian process regression obtained for the target coordinate variable in Scenario II.

Table 2
Tuned hyperparameter obtained using particle swarm optimization in Gaussian process regression for three objects with four coordinate variables each (Scenario II).

Target	Hyperparameter values				
Coordinate variables	θ_0	θ_1	θ_2	θ_3	λ
1	12.359	56.254	51.840	9.275	17.189
2	9.002	50.690	46.310	5.046	60.473
3	0.130	55.443	54.637	−0.380	55.126
4	0.548	24.425	62.118	−0.573	108.492
5	−8.172	94.264	77.497	50.605	57.844
6	9.181	79.162	35.202	14.843	40.920
7	18.989	55.319	88.712	−7.182	67.899
8	−4.958	33.939	58.522	36.850	61.056
9	34.409	−23.239	86.437	5.585	65.577
10	10.041	93.836	31.346	−1.744	48.303
11	0.853	51.439	67.845	15.337	48.158
12	4.256	42.023	47.473	6.650	15.271

predicted coordinates for a typical LiDAR test data sample is obtained using Eqs. (2), (3), (4), and (6). The original coordinate label is 0.33, while the estimated predicted coordinate value is 0.38 for one particular LiDAR test data coordinate. Random noise is considered on the observed target values in the GPR model. where, for different λ values in the target, we apply the Gaussian process [20], where k is the kernel function of the Gaussian process and x_m and x_n are the training datasets. The samples are plotted for various values of $\theta_0 \ldots \theta_3$. To train the network, a matrix of size $(N \times N)$ is formed using the Gaussian process. The covariance matrix (C-matrix) is obtained by adding the noise factor $\frac{1}{\lambda}$ to the diagonal element of the above matrix. To calculate the predicted points, the Gaussian process function matrix "D" of size $(1 \times N)$ is formed with the "M" test vectors and "N" training vectors. The target of the training dataset is used to create the "Target" matrix, which has a size of $N \times 1$. Each target position is now obtained as y_{test}, with the target matrix corresponding to the sample of the first position for the first predicted location. All of the predicted locations are treated similarly. implemented. The PSO convergence graph is a plot that depicts how the algorithm performs over a number of iterations (refer to Fig. 14).

The convergence graph explains how the fitness score, or objective function value, evolves. The graph typically displays the optimal fitness value of the algorithm determined at each generation or iteration.

The objective function value is the fitness value, such as the minimization or maximization, which the PSO attempts to optimize. The graph makes the convergence behavior of PSO easier to understand and gives insights into how quickly or slowly the algorithm moves toward the ideal outcome.

The hyperparameters, θ_0, θ_1, θ_2, θ_3, and λ are tuned using PSO in GPR. The optical values obtained for each target coordinate variable for the three objects (humans) in Scenario II are tabulated in Table 2. The actual time taken to estimate the coordinates of the single test data sample is 0.0043512 s.

Figs. 15, 16, and, 17, of the Scenario II dataset depicts two sample plots showing the locations of the original (acquired using the camera) and predicted target coordinates (using LiDAR).

Fig. 18 shows sample plots illustrating bounding boxes for two randomly selected test data samples. These plots were generated using the GPR technique, with hyperparameters tuned using PSO in Scenario II. The location of the original object coordinates was obtained using the camera, and the predicted object coordinates

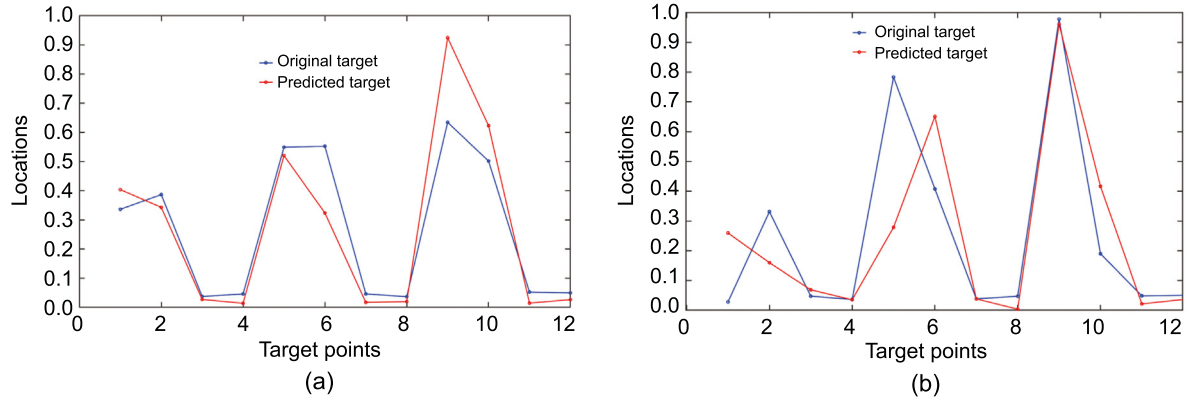

Fig. 15. Comparison of original (acquired using the camera) and CNN-based predicted coordinates using LiDAR data for Scenario II. (a) Sample test data 1. (b) Sample test data 2.

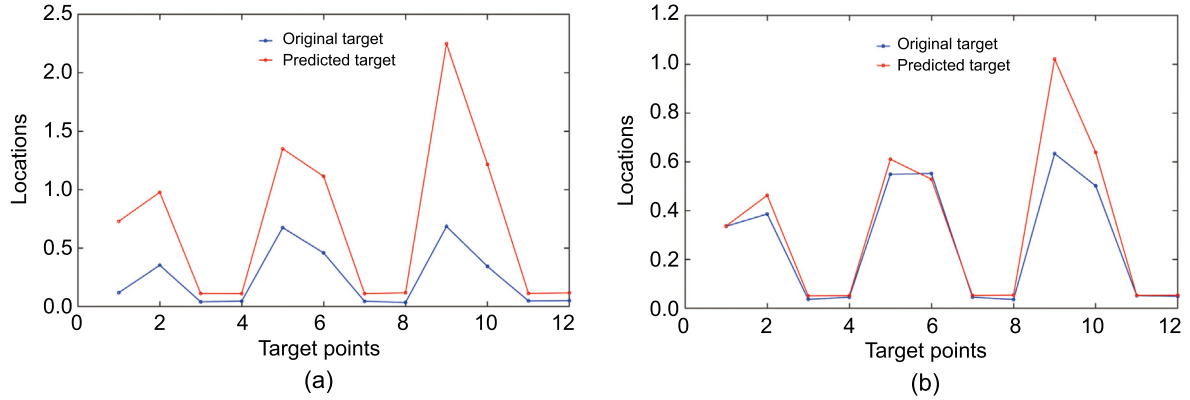

Fig. 16. Comparison of original (acquired using the camera) and Kernel-smoothing-based predicted coordinates using LiDAR data for Scenario II. (a) Sample test data 1. (b) Sample test data 2.

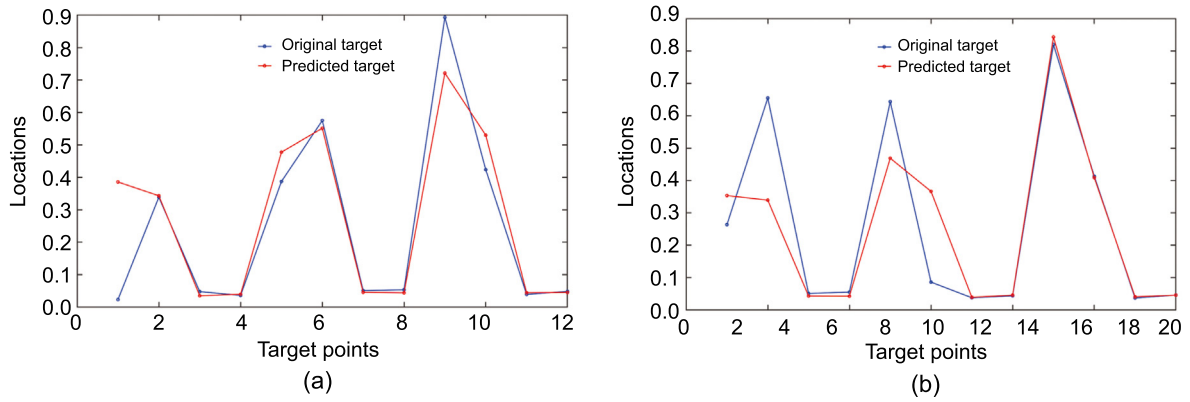

Fig. 17. Comparison of original (acquired using the camera) and GPR-based predicted coordinates using LiDAR data for Scenario II. (a) Sample test data 1. (b) Sample test data 2.

were determined using LiDAR. The three colors (red, blue, and green) indicate the coordinates of the various objects. The circle shows the center point of the object coordinate, and the square indicates the center point of the predicted coordinate by LiDAR.

4.3. Comparison of GPR with KSR and CNN

The metrics used to evaluate the performance of the regression techniques used in this study are mean squared error (MSE), root mean squared error (RMSE), mean absolute percentage error (MAPE), and mean absolute error (MAE) (refer to Table 3). The bold text in the Table 3 represents the lowest error value of the different regression techniques.

Observations (Table 3) indicate that GPR performs better in both scenarios than the other techniques. There is a noticeable improvement in error reduction when the hyperparameters in GPR are tuned using PSO.

In CNN, the Adam (adaptive moment estimation) optimizer is used. The number of epochs is fixed at 20, which implies one iteration per epoch, corresponding to a total number of iterations of 20. The initial learning rate is 0.001 with a minimum batch size of 32, and the L2 regularization is 0.0005 (refer to Fig. 19).

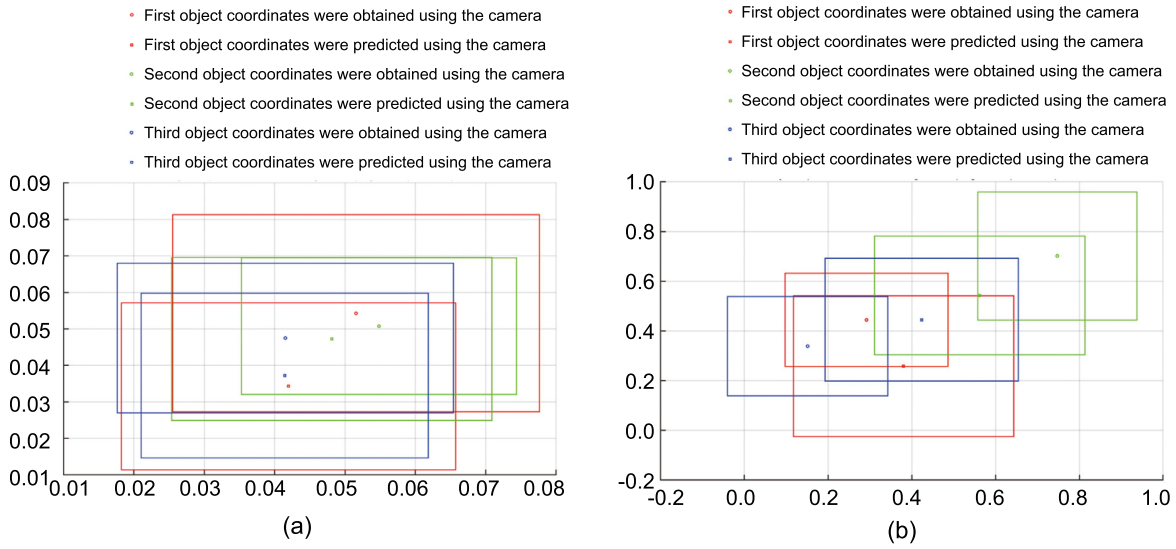

Fig. 18. Comparison of coordinates (center of the ROI corresponding to the three target objects) acquired using the camera (square) and predicted using Gaussian process regression with LiDAR data (circle) for Scenario II. (a) Sample test data 1. (b) Sample test data 2.

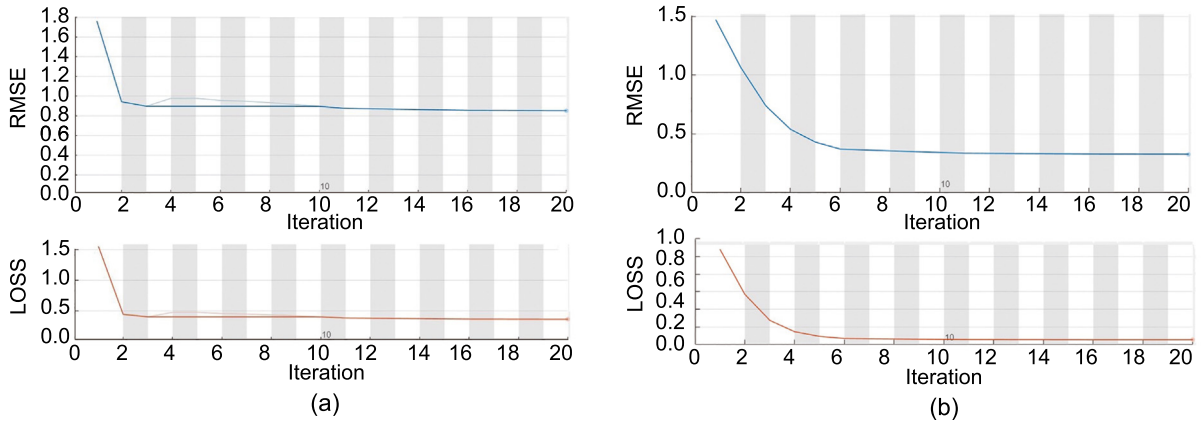

Fig. 19. Convolution neural networks convergence graph attained for (a) scenario I and (b) scenario II.

Table 3
Estimation of error.

Dataset	Regression Techniques	MSE	RMSE	MAPE	MAE
Scenario I	CNN	0.034266	0.034266	1.369061	0.143818
	KSR	0.040602	0.040602	**0.224962**	0.162680
	GPR with random search	**0.014940**	**0.014940**	0.406499	**0.093317**
Scenario II	CNN	0.03368	0.03371	0.69632	0.11467
	KSR	0.11069	0.11069	0.69376	0.1757
	GPR with PSO	**0.01779**	**0.01785**	**0.47313**	**0.07805**

5. Conclusion and future scope

The proposed work shows the importance of using 2D LiDAR for data collection, allowing the collection of detailed environmental information without compromising privacy. Our technique is demonstrated to effectively detect objects within a closed environment by analyzing the collected data and employing regression techniques.

The proposed work provides an experimental analysis of regression techniques used for object detection. The GPR/PSO technique exhibits an error of 0.01, lower than the KSR and CNN techniques. The study demonstrates how to estimate the coordinates of objects using camera target values, paving the way for the replacement of CCTV cameras with LiDAR for surveillance.

In the future, the integration of LiDAR and regression techniques in IoT applications holds significant potential. Combining the capabilities of LiDAR with the power of regression algorithms can enhance various domains, such as autonomous systems, robotics, surveillance systems, and smart-city applications. By leveraging LiDAR's privacy-preserving nature, safer and more efficient IoT solutions can be realized in the future.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H. Lvwen, C. Siyuan, Z. Jianfeng, C. Bang, L. Mingqing, Real-time motion tracking for indoor moving sphere objects with a LiDAR sensor, Sens. Transp. (2017).

[2] B. Drazen, K. Takayuki, I. Tetsushi, M. Takahiro, Person tracking in large public spaces using 3-D range sensors, IEEE Trans. Hum.-Mach. Syst. 43 (2013).

[3] C.H. Alejandra, G. Clara, C. Jonathan, B. Ramón, Object detection applied to indoor environments for mobile robot navigation, Phys. Sens. (2016).

[4] MahmudulHasan, J. Hanawa, R. Goto, R. Suzuki, H. Fukuda, Y. Kuno, Y. Kobayashi, LiDAR-based detection, tracking, and property estimation: A contemporary review, Neurocomputing 506 (2022) 393–405.

[5] Y. Wu, Y. Wang, S. Zhang, H. Ogai, Deep 3D object detection networks using LiDAR data: A review, IEEE Sens. J. 21 (2021) 1152–1171.

[6] S. Demetri, P. Gian, L. Bruzzone, Laps: Lidar-assisted placement of wireless sensor networks in forests, ACM Trans. Sensor Netw. 15 (2019) 1–40.

[7] C. Benedek, A. Majdik, B. sNagy, Z. Rozsa, T. Sziranyi, Positioning and perception in LIDAR point clouds, Digit. Signal Process. 119 (2021).

[8] W. Liu, S. Chen, E. Hauser, Lidar-based bridge structure defect detection, Exper. Tech. 35 (2017) 27–34.

[9] B. Behroozpour, A.M.S. Phillip, M.C. Wu, E.B. Bernhard, Lidar system architectures and circuits, in: IEEE Communications Magazine. Vol. 55, 2017, pp. 135–142.

[10] Y. Yao, S. Fatholahi, Y. Chen, J. Li, Indoor LiDAR point clouds upsampling for object detection enhancement, in: Joint Urban Remote Sensing Event, JURSE, 2023.

[11] R. Wang, M. An, S. Shao, M. Yu, S. Wang, X. Xu, Lidar sensor-based object recognition using machine learning, J. Russ. Laser Res. 42 (2021) 484–493.

[12] F. Bu, T. Le, X. Du, R. Vasudevan, R. Vasudevan, M. Johnson-Roberson., Pedestrian planar LiDAR pose (PPLP) network for oriented pedestrian detection based on planar LiDAR and monocular images, IEEE Robot. Autom. Lett. 5 (2020) 1626–1633.

[13] K. Sohn, A. Bernardi, O.R. Neiroukh, Effects of input resolution on LIDAR pedestrian localization using deep learning, in: International Conference on Electronics, Information, and Communication, ICEIC, 2021.

[14] J. Li, R. Stevenson, 2D LiDAR and camera fusion using motion cues for indoor layout estimation, in: IEEE 24th International Conference on Information Fusion, FUSION, 2021.

[15] H. Miawarni, T.A. Sardjono, E. Setijadi, Wijayanti, D. Arraziqi, A.B. Gumelar, M.H. Purnomo, Fall detection system for elderly based on 2D LiDAR: A preliminary study of fall incident and activities of daily living (ADL) detection, in: International Conference on Computer Engineering, Network, and Intelligent Multimedia, CENIM, 2020.

[16] S. Pintea, S. Karaoğlu, J. van Gemert, A. Smeulders, Large scale Gaussian process for overlap-based object proposal scoring, Comput. Vis. Image Understand. 150 (2016) 95–108.

[17] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, 1995.

[18] E.S. Gopi, Pattern recognition and computational intelligence techniques using matlab, in: Transactions on Computational Science and Computational Intelligence, TRACOSCI, Springer, 2020.

[19] S. Miao, Z.J. Wang, L. Rui, A CNN regression approach for real-time 2D/3D registration, IEEE Trans. Med. Imaging 35 (2016) 1352–1363.

[20] M.B. Christopher, Pattern recognition and machine learning, in: Information Science and Statistics. Vol. 1, Springer New York, NY, 2006, p. XX, 738.