



Trajectory prediction and tracking using a multi-behaviour social particle filter

Vaibhav Malviya¹ · Rahul Kala¹

Accepted: 20 February 2021 / Published online: 21 September 2021
 © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

3D motion tracking is a challenging task when both the tracked object and the observer are moving. In this paper, we present a multi-behavioural social force-based particle filter to track a group of moving humans from a moving robot using a limited field-of-view monocular camera. The application is a robotic guide and while moving, the robot often loses visibility of one or more people in the group, who must still be tracked. As an example, due to limited space, when the robot takes a sharp turn to avoid an obstacle or circumvent a corner, the visibility of the people at the rear is lost for some time. Therefore, several human social behavioural aspects have been implemented to predict the human’s motion in a group. The model accounts for attraction and repulsion between the people of the group and those with the robot, to maintain a comfortable social distance with each other at equilibrium. Additionally, when any person leaves the group then the track is deleted and after joining the track is automatically re-initialized. In the literature, the time of invisibility is a criterion to detect a person who has left the system, which however cannot be used here since the invisibility may be due to a limited field of view or the robot making a sharp turn to avoid an obstacle or circumventing a corner. Social heuristics are used to accurately detect people leaving the robotic system. The tracked trajectory is compared with ground truth and our system gives a very less error when compared with several baseline approaches. False positives are reduced, and the accuracy also increased with our proposed model as compared to other baseline methods. This method has been tested on several scenarios to ensure its validity.

Keywords 3D tracking · Socialistic behaviour · Particle filter · Autonomous navigation · Social robot motion planning · Service robotics · Social potential field

Symbols

Particle filter

$s(x, y)$	Position of the person being tracked	$R_t = (R_t.x, R_t.y, R_t.\theta)$	All observations from time 1 to t
$h_{t,p}^{(i)} = (h_{t,p}^{(i).x}, h_{t,p}^{(i).y})$	Hypothesis of the pose of person p as per the i^{th} particle at time t	$R_{1:t}$	Robot pose (X, Y, orientation) at time t
$w_{t,p}^{(i)}$	Weight of the i^{th} particle tracking pose of person p as per the at time t	$q_{1:t,p}$	Robot pose from time 1 to t
$Z_{t,p}$	Observation of person p at time t	$S_{1:t} = \{q_{1:t,p}, R_{1:t}\}$	Pose of all people from time 1 to t
$Z_{1:t}$		$P(h_{t,p} Z_{1:t}, S_{1:t})$	System state $S_{1:t}$ consisting of all person’s state $q_{1:t,p}$ and the pose (state) of the robot $R_{1:t}$ from time 1 to t
		$p(Z_{t,p} h_{t,p}^{(i)})$	Probability density function while tracking person p at time t
		$\hat{k}_{t,p}$	The likelihood function or probability of observing $Z_{t,p}$ if the person is at $h_{t,p}^{(i)}$
		n	Set of all weighted particles at time t tracking person p
		$\mathcal{U}_{t,p}^{(i)} = \text{anticipate}(h_{t,p}^{(i)}, S_t)$	The number of the particles

✉ Vaibhav Malviya
 vaibsidea@gmail.com

Rahul Kala
 rkala001@gmail.com

¹ Centre of Intelligent Robotics, Department of Information Technology, Indian Institute of Information Technology, Allahabad, Prayagraj, India

$k(h_{t,p}, U_{t,p})$	The guessed control input for person p hypothesized by particle i at time t	$W_t \in [0, 1]$	Relative importance/weight of social forces
\int_{noise}	The kinematic equation of motion	W_{th} and σ_{th}	Mean and standard deviation of selecting W_t when the person is visible
c	Motion noise used for uncertainty in the motion model	W_{min} and W_{max}	Range of selecting W_t when the person is not visible
$q_{t,p} = (q_{t,p}.x, q_{t,p}.y)$	Correspondance function (tells which observed person is which person in the database)	$\vartheta(h_{t,p}^{(i)})$	The linear speed of i^{th} particle for human p at time t
Motion model	The current tracked state of the person p	v_{max}	Maximum velocity of the person
$f_{attr}(R_t, h_{t,p}^{(i)})$	Attraction force between the person represented by the particle $h_{t,p}^{(i)}$ and the robot at R_t	$\theta(h_{t,p}^{(i)})$	Desired angle of motion for person p as per the i^{th} particle
$f_{rep}(R_t, h_{t,p}^{(i)})$	Repulsion force between the person represented by the particle $h_{t,p}^{(i)}$ and the robot at R_t	δ_t	The time between the consecutive iterations or the inverse of the tracking frequency
$f_{attr}(q_{t,p'}, h_{t,p}^{(i)})$	The attraction force between the person represented by the particle $h_{t,p}^{(i)}$ and another person p' at $q_{t,p'}$	Observation model	
$f_{rep}(q_{t,p'}, h_{t,p}^{(i)})$	The repulsion force between the person represented by the particle $h_{t,p}^{(i)}$ and another person p' at $q_{t,p'}$	C	Calibration matrix
$f_{rand}(h_{t,p}^{(i)})$	Random force	f_x, f_y	Focal length of the camera
$f_{total}(h_{t,p}^{(i)})$	Total social force for i^{th} particle of the person p at time t	(φ_x, φ_y)	The principal point offset of the camera
$= (f_{total}(h_{t,p}^{(i)}) \cdot x, f_{total}(h_{t,p}^{(i)}) \cdot y)$		β_s	Axis skew or shear distortion in the projected image
w_1, w_2, w_3, w_4, w_5	Constants that denote the relative importance of all above forces	k_c	Height of the camera in the Z-plane
$d(a, b)$	Distance between a and b	$T_{camera'}^W$	The transformation between the camera' and the world coordinate axis
$\alpha(b, a)$	Orientation between a and b	$T_{camera}^{camera'}$	The transformation between the camera' and camera coordinate frame
$\rho(R)$	The radius of the robot	$(u_{t,p}^{(i)}, v_{t,p}^{(i)})$	Expected position in the image plane for the person p at time t as per particle i .
$\rho(p)$	The radius of the person p	$z_{t,p} = (u_{t,p}, v_{t,p})$	Actual observed position in the image plane for the person p at time t
σ	Constant used to determine how quickly the repulsive force with robot fades with an increase of the distance	H	Height of the image
σ'	Constant used to determine how quickly the repulsive force with people fades with an increase of the distance	η	Normalization factor
$r \sim U[a, b]$	A random sample r taken from a uniform distribution between a and b	$visible(t)$	Whether the person was detected at time t or not
$r \sim N[a, b]$	A random sample r taken from a normal distribution of mean a and standard deviation b	$left(t)$	A function to find whether the person has left the group or not
		Δ	Threshold of time within which if the person is not seen, the person is said to have left the group
		$G_t = (G_t.X, G_t.Y)$	The sub-goal that the robot is currently seeking
		$turn(t)$	Function to determine whether the robot is making a sharp turn or not
		$\alpha_e(t)$	Robot orientation towards the sub-goal
		δ	The maximum angular deviation in the natural navigation of the robot
		Acronyms	
		TI	Tracking in Image
		RPF	Random Particle Filter
		KF	Kalman Filter
		LSTM (wr)	

	Long Short Term Memory (with re-initialization)
LSTM (w/o r)	Long Short Term Memory (without re-initialization)
3D-PSFM	3D Particle Social Force Model
RMSE	Root Mean Squared Error
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
P	Precision
R	Recall
A	Accuracy
F	F-measure / F-score

1 Introduction

With the development of the technology in the area of the robotics, the robots are now seen in every field such as medical robotics, robots as an assistant in home and office environments, robots to take care of the specially-abled, etc. A social robot is used to navigate along with humans for multiple tasks such as a guide at tourist places, assistants in stores and shopping malls, carrying human subjects in hospitals for X-ray, pathology, MRI, etc. In these navigation scenarios a lot of changes occur such as illumination changes, humans sometimes need to come close to the robot due to limited space, the humans and robot may both need to make sharp changes due to a very less breadth of the corridor (also known as the problem of the narrow corridor), etc. In all such cases, accurate detection of the human is not possible everywhere and so we have to apply human tracking to overcome these situations.

The major application of focus here is a robotic guide. The robot takes a human group on a tour. The robot stops at its goal and explains the relevant facts of that site to the visitors. The robot does not need to stop somewhere if relevant facts might be complete in few words. The complete application was demonstrated in an earlier work of the authors [1]. The work was based on a heuristic that the robot should continue the motion till all people are visible, while in case of invisibility the robot was instructed to first slow down and then stop. The long-term aim is to make the robot interact personally with the people (that requires their accurate position), affirming the absence of the person by turning (for which an indicative position is required), deciding the segments in which the persons need to be mandatorily present and segments where not all humans may be required, etc. Tracking is the hardest challenge in all these aspects, which is centrally taken as a problem in this paper. In such a context, the prior work faces two problems. The first problem is that the noise and false negatives make it impossible for a robot to know the position of a human to face it, for which tracking the solution.

Secondly, the robot's navigation decision is also a function of the persons not visible, for which again tracking is a solution. Previously the authors also performed tracking from a stationary camera to track multiple entities [19].

Many techniques exist to anticipate or predict the motion of a person, based on the knowledge of the other people, robots, and the task that the person is performing. As a naïve implementation, the person may be expected to be moving with the same speeds as observed previously. Accurately predicting the immediate motion of the persons has a lot of applications in social robotics, as the social robot can adapt its behaviour with knowledge of the anticipated behaviour of the neighbouring persons. A motion model is a generic term used to predict the future state of a system, with a knowledge of the current state and sometimes with additional knowledge of the control input to the system. Knowledge of the intent of the person indicates the normal motion that the person should perform. However, a knowledge of the static and dynamic obstacles also indicates the hindrances that must be accounted for. The person may not have a clear intent but either of many intents may be stochastically possible, which also needs to be accounted for. The system, or some of its parts, may even be observable. Like the robot may have anticipated the person's future position, but with time it can also observe the position using the onboard cameras. This constitutes the observation that is often used to correct the predictions. An amalgamation of both motion and observation is called the problem of tracking.

In recent years, 3D human tracking in indoor environments has become a research hotspot in the domain of computer vision and social robotics. However, 3D tracking from a mobile robot becomes challenging when both the robot and humans are moving. It is vital to predicting the accurate position of the human and robot's target continually for different actions. Hence, to track the human in the challenging environments, it is required to localize the human by the robot which differentiates that human's target from the other objects. One of the competing approaches to ours is by Wang et al. [2], who proposed a method to track the human position in 3D by an ultrasonic sensor and a monocular camera in indoor and outdoor scenarios. Here, a real-time visual tracking approach was shown to handle various situations like object re-detection, missing objects, and occlusion. The monocular camera sensor tracking model was installed on the top of the ultrasonic sensors and fixed on the robot. The visual tracking method was based on the Kernelized Correlation Filter [3] tracker. The Ultrasonic sensor module was introduced for tracking where the sonar sensor array was considered to obtain the range information and improved the predicted accuracy of the target. For multi-sensor fusion, the Extended Kalman Filter was applied to fuse the measurements which were obtained from the vision and ultrasonic sensors.

In our application, the humans in a social group also follow the robot while having an affinity towards each other, which

makes the tracking hard. The motion module of the tracker may predict that the humans continuously move towards the robot, however, it may not be possible due to a larger number of people needing more personal space. Similarly, people often form groups within themselves and move as a cluster. This social attraction between human beings brings erratic trends that the tracker must accept. Such social conventions have not been considered in the domain of social tracking of a group of humans in the literature and the proposed work models all such factors for accurate tracking of the humans in a group.

The robot used for the work uses a limited field-of-view monocular camera for detection and tracking of the humans in a group. Therefore, it is evident that humans often get outside the field of view of the camera, although they are still in the group following the robot. Consider the example that the robot is navigating through a narrow corridor and needs to make a sharp turn in the corridor, or to avoid an oncoming obstacle. In such a case, the humans in the group will not be visible to the robot for a prolonged amount of time while the robot is circumventing the sharp turn. Furthermore, the humans following the robot will have to make the same turn, during which also they will not be within the field of view of the robot. Unlike some literature using multiple cameras or robots, here for budgetary reasons only one robot is available to guide the human group and to track them continuously, so the robot must perform both jobs simultaneously without any erroneous. The robot must thus be able to predict the motion of the humans and make navigation decisions autonomously.

Our proposed work is based on 3D tracking where the humans are in a group and follow the robot. The robot knows only the initial position of the humans and tracks them continuously. Since the detection of the human is not possible everywhere due to illumination changes and the need to take sharp turns, so the tracking methodology has been developed to overcome these problems. In this paper, the multi-behaviours 3D particle social force model is proposed to solve this problem which includes various socialistic forces like attraction and repulsion between humans to humans and between humans to the robot. If the humans are far from the robot while following, then the social attraction force is active to know the further position of the humans in the 3D real world. While the human is following the robot, they can also come too close to the robot, in which case the humans are repelled by the robot. Similarly, people may neither be too far or too close to each other. The socialistic tracking is based on the principles that all the entities shall come at a social equilibrium and attempt to maintain the same equilibrium. The equilibrium conditions were derived from a set of social experiments with human subjects.

Through the navigation process of the robot, the human can also behave erratically, like they can interchange their position with someone in a group, one human overtakes or crosses another one, and it is a possibility that human may be beyond

the robot while following it. Specifically, one additional behaviour can occur that humans do not follow the robot and still stand at their previous state to look at that place for a longer time. These social aspects are hard to model as their triggering conditions are non-observable. The tracker should be able to handle such situations still.

Many times, the human can get any telephonic call and to attend it, the human must leave the group. If one human leaves the group, another human may also leave the group due to the conformity of each other. Often it happens that a human can leave the group forever because they might get some emergency call. Our proposed model detects such situations and deletes the tracker if any human leaves the group and as the human re-joins his group, the tracker is automatically determined by computing the visitors' position in the 3D real world and tracking them forward.

In the literature, the problem of deleting tracks is solved largely by the time for which the person is not visible. If a human is not detected for a prolonged time, it is assumed that the human has left, and the track is deleted. In our application, while the robot is moving and navigating the human there may be several sharp turns and corners. So due to a lack of space behind the robot, humans may not exactly stand behind the robot and may be out of the field of view of the camera. Here the human cannot be detected by the robot. So, the time of invisibility is not a suitable metric. In this paper, we detect the possible intents of the robot and the person. If there is enough evidence that the person can follow the robot and the person is not doing the same, only then the track is deleted. This saves a lot of false positives that can have severe advantages from an application perspective.

If the human leaves the group and re-joins the group, then the tracker is also deleted and re-initialized. In the initial phase, the robot is assigned the initial position of the human, and when the tracker is re-initialized, the robot detects the human in the 2D image plane and then converts this 2D image plane coordinate into a 3D real-world coordinate, following which a continuous tracking is performed. In this approach, we have used a monocular camera only to track the human and to determine the position of the human in the 3D real-world. The 3D position can be estimated with knowledge of the 2D position in the image with the assumption that the height of the person is known.

The recent works and development in social robot motion planning in navigation have not substantially investigated the behaviour where humans follow a leader robot, while the works are largely centralized on socially avoiding the humans by a robot or a robot following humans. The crucial challenge that isolates this problem from the rest of the variants is that humans can show non-cooperative and strange behaviours like missing out at times from their group for long durations and the robot must still socially accommodate the behaviour. The robot must understand human behaviour in a group when the robot is navigating. Here humans perform different socialistic behaviour such as leaving the group, swapping their

position with each other, overtaking each other, unnecessary wait at one place for a certain time, etc. The robot must handle these situations and continuously track the humans without any position loss. The other major challenge is that such implementations have a huge number of corner cases considering the limited field of view of the robot. The humans' visibility gets lost when the robot is making a sharp turn at a corner and human detection is not possible. Consider the example of a robot turning at a sharp corner. The human will be out of view of the camera as the robot makes the sharp turn, and if the robot assumes that the human is missing, it will end up waiting at a sharp corner and thus causing a blockage. The humans are found at the back-side, while the robot navigates in the forward direction. This is not a natural human behaviour, and therefore the general heuristics of the social robot motion planning applications fail to recognize the difference in needs in the modelling of this behaviour. So, we propose a novel model based on social force and particle filter which can overcome the above-discussed noisy human behaviour and track humans at the corner where detection is not possible. Beside this, only a less expensive monocular camera (like a webcam) is sufficient to perform 3D tracking in real-time. High expensive cameras (such as stereo cameras, lidar, etc.) are not needed for our proposed model.

Our proposed model has been tested on three different scenarios to check its validity. For verification, we have recorded the actual subject's trajectory in 3D in each scenario using a simulation and compared it from the anticipated human trajectory. Experiments confirm that the proposed methodology detects people leaving the group more accurately and tracks the people more accurately as compared to several baseline approaches. The differences are particularly large when the robot has a limited field of view of the subjects for a long time within the experiment.

The objectives of the paper are:

- To track multiple humans from a moving mobile base
- To track the humans from a limited field of view camera, wherein it is eminent that the humans will often be outside the field of view of the camera.
- To handle the noises arising out of the low precision camera.
- To handle the cases when the person temporarily leaves and re-joins the group, while tracking should not happen during this time.
- To accurately predict whether the loss of visibility is due to the person unable to come within the visible region (in which case the tracking must continue) or due to the person leaving the group (in which case the tracking must discontinue).
- To re-initialize tracking when the person re-joins the group.

We propose a strategy for 3D socialistic tracking in a real-time environment. The main contributions are:

- i. The paper studies a new paradigm of social robotics, where multi-human tracking is performed from a budget and limited field of view monocular camera. This enables using social robots at a budget, while the robot can still make decisions based on the tracked location of the humans. The challenge lies because humans will often get outside the field of view of the camera.
- ii. The system can detect persons leaving the group based on heuristics that incorporate the current context and intends. When any human leaves the group the track is automatically deleted and if the person re-joins, the track is also re-initialized, and tracking continues.
- iii. A socialistic human following behaviour is developed that accounts for social attraction and repulsion forces between the people and those with the robot. The model can predict the motion of the person even with a lack of visibility. The force constants are obtained realistically using real-life socialistic experiments with human subjects.
- iv. The robot can handle the loss of visibility due to sharp turns and corners where detection is not possible attributed to the strong social prediction model. Further, the robot anticipates such a loss of visibility is contextual and the track is not deleted.
- v. The proposed approach more accurately detects people leaving the group and performs a better socialistic tracking when compared to several baseline approaches.

2 Related work

This section lists some of the recent related works from the literature. Our work focuses on the development of a 3D human motion tracking framework where a robot is navigating with humans in a group.

Humans are astute agents and behave accordingly to decision-based strategies. Here the social robot guides humans in a cluttered and dense pedestrian environment. So, the robot can freeze itself [4] if the path is not safe and the probability of collision is high. In [5], the latest phenomenon was proposed for socially adaptive path planning in a non-static environment which included the inverse reinforcement learning, feature extraction, and path planning modules. Furthermore, in [6], the authors developed a new approach that allowed a social robot to learn a model of navigation behaviour of the pedestrians and this approach used the Hamiltonian Markov chain Monte Carlo sampling. In a more advancement and achievement of the socially-aware navigation, a new algorithm was proposed called the SocioSense which was based on the social

constraints [7]. In a recent paper [33], we implemented a hybrid framework to avoid humans according to social conventions followed by the humans, keeping a socialistic distance while avoiding humans. Even though a lot of socialistic studies on human navigation are done, the studies on the specific case of human motion in a group while following a robot is sparse. The research, therefore, models the behaviours and implements the same for tracking.

The robots can be applied to different servicing areas if they are reliable and safely follow humans. The authors in [8] addressed the tracking of humans in a cluttered and dynamic scenario where the tracking algorithm considered the interdependence among the targets. The design module of this tracking was a grouped target which additionally consisted of a leg-pair target. So, the grouped target and target was considered as a human and human leg. When the human legs are observed from the LRF (Laser Range Finder), then the targets (human legs) are not independent. Specifically, in a crowded and cluttered environment, the human has to be sorted among the measurements which are observed. Here the authors developed HSJPDFAF (hierarchical sample-based joint probabilistic data association filter) for robust tracking, which was able to predict the state of the target corresponding to their legs and could also find the positions of the human. HSJPDFAF was divided into two levels, low and high. At a low level, the observation model was used to predict the state of targets legs while at the high level, the states of group target id were determined with the association of the observation model. The robot was activated to move towards the human if the target of humans was available via a data association. In contrast, our approach focuses on cases where there is no detection at all for parts of the tracking process, which can only be corrected by accurate socialistic modelling of human behaviour.

A mobile robot should be designed with a system capable of detecting the target person, integrating different types of surrounding information, and finding the optimal control scheme. The human following framework consists of three attributes: human detection, human recognition, and human tracking [10]. In these circumstances, target recognition is a challenging task for human following. In [11], the principal component analysis was combined with tracking learning detection and applied on the human face (Face-TLD) to achieve an improvement which is called IFace-TLD. The IFace-TLD improved the discrimination ability of the Face-TLD, like ambiguous facial appearances. Motion uncertainty was dealt with by a skeleton-based model which improved the robustness and accuracy of the target recognition. A particle filter was used to achieve the human following system. The particle filter predicted the human motion state. A controller was modelled to maintain a relative distance between the target and the robot, and this controller made a satisfactory performance for the human following. Similarly, head pose estimation [18] was utilized for 3D tracking pose estimation. In [19] authors performed a face tracking approach with

different height of the camera that can be applied to know the best field of view of the camera. A multitask convolution network [21, 22] was applied for face detection and alignment. Our work instead models group behaviours where a group of people follows a robot, including an accurate detection of people leaving the group and tracking even in low-field-of-view conditions. The group leaving detection is primarily driven by an understanding of the context of the robot operation.

Trajectory prediction is a problem related to tracking. A lot of recent works use deep learning models to predict the trajectory of several human agents. In [29] a 3D network model was developed for the trajectory prediction using LSTM which was based on the human pose. In an unconstrained environment, a CNN model [30] was proposed for human detection and tracking. Additionally, memory augmented network [32] has been modelled and utilized for multiple trajectory predictions on the road and side-walk with the injection of CNN encoder. The problem with the methods is the need for a large quantity of noise-free training data, that is not available as per our modelling. We assume that the human will not always be visible, the human intents will be highly noisy, and the captured image of the humans will also be highly noisy. This makes it impossible to learn the human behaviours and we instead use heuristic social potential field methods.

Some traditional methods have also been used for prediction and tracking. In [31], Physics and maneuver based model was utilized with the integration of unscented Kalman filter and Monte-Carlo method for the prediction of the vehicle trajectory. Moreover, a vehicle kinematic model and driving behaviour awareness network have been developed. In a paper [34], a fuzzy neural network with the kinematic model and parameter learning was used for the tracking of a robot trajectory. In a real-time scenario, pedestrian model-based reactive planning was performed for assistive robots [35]. Human pose clustering over multiple motion sequences and its encoding has been applied for tracking data to visualize the motion flow [36]. Here motion pattern was also applicable to predict the split, merge, and lock situation of tracking data. Silhouette image and edge detection are widely used for human motion tracking. In [37] a filtering algorithm with stereo calibration was applied with the Markov model to perform human tracking. Further, in a paper [38], human detection and tracking were performed in a distinct environment. These approaches solve the problems associated with correspondence, detection, and handling noises during tracking. Our challenge is amplified since the data is recorded from a moving base and it records another moving object, and further the limited field of view cameras make the human often lie outside the visibility of the cameras. These are new challenges not accounted for in the related works.

Many general robot behaviours have been studied in the literature, especially in the planning context. In [12], a mechanism was proposed for path planning which included the reinforcement learning, and path planning modules for a

mobile robot. In [13], a real-time 3D trajectory planner was proposed in the context of an autonomous vehicle for leader following. Here the planner was based on the basic concept of trailer body whose hinge point was attached with the leader and every follower was assigned to a different point of the trailer. The Lyapunov analysis was used to represent the trailer body reference frame which ensured that the planning could be separately designed by each follower. So, the application of this designing system made the need for communication among followers negligible. Trajectory planning and trajectory tracking were the main steps of this phenomenon. Furthermore, the trajectory planner generated a natural trajectory for the followers which was not similar to the trajectory of a leader. When multiple robots move and behave like a swarm group, then they can follow a leader robot in a straight line and sort themselves to stand in a line according to alpha-numeric name (identity) as demonstrated in [14]. In [17], a framework was developed for robotic exploration in a given static environment. In the development of swarm robotics [20], particle swarm optimization, and fruit fly optimization algorithm are also used. A related domain of work is the motion planning of the robots. In [26], mathematical modelling and construction of the potential field function for the obstacle avoidance and dynamic modelling of the car were done. Our approach adds social primitives in the potential field algorithm to track humans. Furthermore, the robot motion model based on the artificial potential field by adding deliberation to reactive algorithms is an efficient method for indoor path planning [15]. In the context of the social robot navigation, the group split and again merge and this thing can be predicted by the 3D convolution neural network [16].

Social robotics is attaining interest from different application perspectives. In [27], a medical application was modelled onto the serving robot that can assist in surgery. The authors also produced reliable and fast control signal which can deal with different changing environmental conditions. The robot can also be employed in medical operation and surgeries. The movement of the robot is an essential task for doing a successful job. It is also called as Remote Center of Motion (RCM) [9]. The authors used a combination of RCM and path following controller. In general, the robot goes into the patient body via incision point and performs its surgeries. Here a surgical robot rotates along the three axes (X, Y, Z) on the penetration point, but it is also limited to translate in one direction. To perform such type of task with the robot, a special kinematic structure design is required. For the robot motion control, two kinds of controllers are needed for navigating the manipulator over a reference curve, either a trajectory tracking, or path following. Here trajectory tracking was considered as time-dependent, where the reference curve was parameterized with the time. The path following behaviour was time-invariant, where the curve was interpreted without any temporal constraints. RCM issue was introduced to implement control software for performing fulcrum tasks along with

time-invariant 3D path following which was based on a visual controller. Here different reference frames were used. The two-dimension motion of the tool (linear) was restricted, and only one-dimension translation was allowed. In our experiments, the robotic leader needs to follow such a trajectory that for similar reasons is filled with sharp and non-ideal turns.

There are other broad problems often discussed related to social robotics and navigation. In a development context of the social robotics application, in [28] an emotional classification was performed using the ECG, GSR, and BA sensors. Moreover, a hand gesture segmentation approach [24] has been developed for feature fusion in a complex background. Beyond this fuzzy logic [23], text detection and tracking [25], can also be applied for navigation and notification. Our application instead is to make the robot lead a human group using a limited field of view sensors, which has its different challenges.

3 Particle filter

In this paper, a 3D particle filter is used for multi-human tracking. Here a set of particles or samples is used to represent the probability density function. Each particle represents the state variable with some numerical value. It is an efficient and alternative method to demonstrate and keep the probability density function non-stationary and non-linear. Here the robot tracks humans who are moving in a group and thus the pose is given by Eq. (1)

$$s = \begin{pmatrix} x \\ y \end{pmatrix} \quad (1)$$

where (x, y) is the 2D coordinate of the person.

In the particle filter [39, 40], the posterior density function $P(h_{t,p} | Z_{1:t}, S_{1:t})$ is given by all the previous observations $z_{1:t}$ and an unknown control signal which is ultimately derived from every previous positions of the robot ($R_{1:t}$) and other people ($q_{1:t,p}$), together making the system state $S_{1:t} = \{q_{1:t,p}, R_{1:t}\}$. The subscript t is for time and p is for the person being tracked. The particle filter is defined (in Eq. (2)) as a set of weighted particles.

$$k_{t,p} = \left\{ \left(h_{t,p}^{(1)}, w_{t,p}^{(1)} \right), \left(h_{t,p}^{(2)}, w_{t,p}^{(2)} \right), \dots, \left(h_{t,p}^{(n)}, w_{t,p}^{(n)} \right) \right\} \quad (2)$$

In Eq. (2), $h_{t,p}^{(i)}$ represents the i^{th} particle sample which is the position or state of the human p at time t . Here subscript t is used for a time, p for the person being tracked, and n denotes the number of the particles. $w_{t,p}^{(1)}$ is the corresponding weight of the particle.

The first step in the implementation of a particle filter is to initialize the particles based on the known state of the system. In our case, it is assumed that the initial position of the persons is known and therefore the particles are spread around the

known initial positions based on the knowledge of the noise in measuring the initial positions. The initial weights are assumed to be the same for all particles. The filter continuously updates the positions and weights of the particles.

The filter applies the motion model to update the knowledge of the state of the system. Our 3D motion model is based on multiple social forces and anticipates the position of every human in the group. The problem is, given the positions of all people and the robot, the system must anticipate the control input that the person will invoke given by Eq. (3) and the update state of the person because of the control given by Eq. (4).

$$\mathcal{U}_{t,p}^{(i)} = \text{anticipate} \left(h_{t,p}^{(i)}, S_t \right) + \mathcal{Y}_{noise} \quad (3)$$

$$h_{t+1,p}^{(i)} = \mathcal{K} \left(h_{t,p}^{(i)}, \mathcal{U}_{t,p}^{(i)} \right) \quad (4)$$

Together Eq. (3–4) represent the motion model of every particle. In the above expression, $\mathcal{U}_{t,p}^{(i)}$ is the anticipated control for humans which consists of the linear and angular speeds. Here function “anticipate” tries to capture humans’ motion on a given current state $h_{t,p}^{(i)}$ corresponding to the i^{th} particle. Moreover, \mathcal{Y}_{noise} is the motion noise used for uncertainty in model anticipation and \mathcal{K} is the kinematic equation of motion.

The particle filter then takes an observation $Z_{t,c(o)}$. Here c is the correspondence function that maps the o^{th} observation to the corresponding person $p = c(o)$ and t is the time. The observation is used to weight the particles. The weight ($w_{t,p}^{(i)}$) is directly proportional to the likelihood function $p \left(Z_{t,c(o)} | h_{t,p}^{(i)} \right)$. This likelihood function is the probability of observing $Z_{t,p}$, given the current state as $h_{t,p}^{(i)}$.

The implementation of such a particle filter will soon have many particles in poor areas. Hence a re-sampling is done where the particles at good areas with high weights are replicated many times, while the ones at poor areas with low weight values die off, representing a new particle distribution with more particles at good areas.

In our proposed model of the 3D particle filter, the current state $q_{t,p}$ of the human p is anticipated by the weighted mean of the particle as expressed in Eq. (5)

$$q_{t,p} = \sum_{i=1}^n w_{t,p}^{(i)} h_{t,p}^{(i)} \quad (5)$$

4 Design philosophy

Section 3 discussed an overall approach of using a particle filter to solve any tracking problem. This section aims to solve the problem as per the stated objectives using a particle filter. The implementation of the particle filter specifically asks for a motion

model to first anticipate the motion of the person hypothesized described in a generic sense by using Eq. (4) and then to move the particle using the same anticipated model described in a generic sense by using Eq. (5). The motion model has a high value for our work since the person will often be outside the visible region and the accuracy of the system largely depends upon the motion model to accurately predict the motion of all persons. The motion model discussed in Sec. 4.1 uses the concept of a social potential field that measures how much a person is attracted and repelled by the other people in the vicinity and the robot to predict the person’s motion.

The next important aspect of a Particle Filter is to correct the errors developed in the motion model and is done by using an observation model. Every particle mentions a hypothesised position of the person. The observation model projects the hypothesized positions of the persons onto the image taken by the camera, given the location of the robot and hence the location of the camera. If the hypothesis was good, the projected position of the person will match the observed position on the image and vice versa. This is used to re-weight the particles. After re-weighting the particles are re-sampled that increases the particle density at good areas and decreases the particle density in poorer areas.

We have developed a framework using multi-behaviour 3D particle social force model which consists of human detection, analysis of human behaviour, and different force applied to compute human position in 3D real-world and 2D image plane. The different modules are discussed in the following sub-sections.

4.1 Motion model

The purpose of the motion model is to predict the control input $\mathcal{U}_{t,p}^{(i)}$ for particle i and person p at time t , given the state of the system is being continuously tracked. The person may not be visible for a long time and therefore the onus of correctly tracking the person relies upon the motion model alone. The motion model is made using the social potential field algorithm, where essentially every particle witnesses a social force from different factors that govern the overall motion of the particle. A particle representing a person’s position feels an attraction towards the robot that is dominant when the person is far away. The person also feels a repulsion on being too close to the robot. Similarly, the person feels an attraction from the other people in the vicinity on being far away. This behaviour makes the group stay intact. Similarly, the people feel repulsion between each other on being too close, which is also often observed as the people like to have some personal space. The particle is moved based on the cumulative effect of all these forces. This model is used to anticipate the state of the person at the next time instant. A representation of the notations is shown in Fig. 1.

The first force is the attraction between the person represented by the particle $h_{t,p}^{(i)}$ and the robot whose accurate pose R_t is known

by the onboard localization system. An only attraction may make the person come too close to the robot, almost kissing the surface of the robot, which is not practical. Therefore, an additional repulsive force is modelled that becomes dominant when the robot is too close. Together the attraction and repulsive forces between the person and robot make the person follow the robot maintaining a comfortable socialistic distance. The attraction and repulsion force between the robot and human is computed as in Eq. (6) and Eq. (7) respectively

$$\overrightarrow{f_{attr}(R_t, h_{t,p}^{(i)})} = w_1 d(R_t, h_{t,p}^{(i)}) \begin{bmatrix} \cos\alpha(R_t, h_{t,p}^{(i)}) \\ \sin\alpha(R_t, h_{t,p}^{(i)}) \end{bmatrix} \quad (6)$$

$$\overrightarrow{f_{rep}(R_t, h_{t,p}^{(i)})} = -w_2 \exp\left(\frac{-d^2(R_t, h_{t,p}^{(i)})}{2\sigma^2}\right) \begin{bmatrix} \cos\alpha(R_t, h_{t,p}^{(i)}) \\ \sin\alpha(R_t, h_{t,p}^{(i)}) \end{bmatrix} \quad (7)$$

w_1 and w_2 are constants that denote the relative importance of the two forces. σ is another constant that determines how quickly the repulsive force fades with an increase of the distance. Here the distance between the particle $h_{t,p}^{(i)}$ and robot R is given by Eq. (8)

$$d(R_t, h_{t,p}^{(i)}) = \sqrt{(R_t.x - h_{t,p}^{(i)}.x)^2 + (R_t.y - h_{t,p}^{(i)}.y)^2} - (\rho(R) + \rho(p)) \quad (8)$$

$\rho(R)$ is the radius of the robot and $\rho(p)$ is the radius of the human p . $\alpha(R_t, h_{t,p}^{(i)})$ is orientation between the robot and human, given by Eq. (9)

$$\alpha(R_t, h_{t,p}^{(i)}) = \text{atan2}(R_t.y - h_{t,p}^{(i)}.y, R_t.x - h_{t,p}^{(i)}.x) \quad (9)$$

For larger groups, it is further observed that every person in the group cannot follow the robot as there is little space and people cannot overlap. Furthermore, people tend to keep a comfortable space between themselves even though they may not be towards a collision. Many times people stick together as a group due to an attraction between themselves. In such a case a person may not only follow the robot but also wait for lagging members of the group. Therefore, there is repulsion and attraction between every two people in the group. The attraction and repulsive force among every human in the group is given by Eq. (10–11)

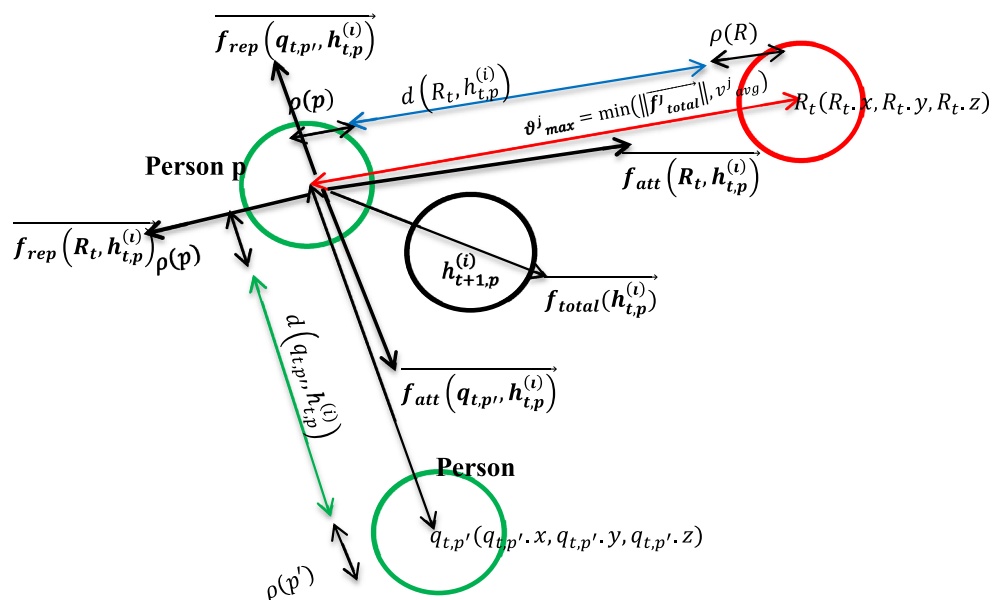
$$\overrightarrow{f_{attr}(q_{t,p'}, h_{t,p}^{(i)})} = w_3 d(q_{t,p'}, h_{t,p}^{(i)}) \begin{bmatrix} \cos\alpha(q_{t,p'}, h_{t,p}^{(i)}) \\ \sin\alpha(q_{t,p'}, h_{t,p}^{(i)}) \end{bmatrix} \quad (10)$$

$$\overrightarrow{f_{rep}(q_{t,p'}, h_{t,p}^{(i)})} = -w_4 \exp\left(\frac{-d^2(q_{t,p'}, h_{t,p}^{(i)})}{2\sigma'^2}\right) \begin{bmatrix} \cos\alpha(q_{t,p'}, h_{t,p}^{(i)}) \\ \sin\alpha(q_{t,p'}, h_{t,p}^{(i)}) \end{bmatrix} \quad (11)$$

w_3 and w_4 are the proportionality constant denoting the importance of the force. σ' determines how quickly the repulsive force fades with an increase of the distance. $d(q_{t,p'}, h_{t,p}^{(i)})$ and $\alpha(q_{t,p'}, h_{t,p}^{(i)})$ are distance and orientation between the person p' (as tracked by the system) to the human being tracked represented by the particle $h_{t,p}^{(i)}$, given by Eq. (12–13)

$$d(q_{t,p'}, h_{t,p}^{(i)}) = \sqrt{(q_{t,p'}.x - h_{t,p}^{(i)}.x)^2 + (q_{t,p'}.y - h_{t,p}^{(i)}.y)^2} - (\rho(p) + \rho(p')) \quad (12)$$

Fig. 1 Multi-behaviour based computational geometry of 3D-PSFM



$$\alpha(q_{t,p'}, h_{t,p}^{(i)}) = \text{atan2}(q_{t,p'} \cdot y - h_{t,p}^{(i)} \cdot y, q_{t,p'} \cdot x - h_{t,p}^{(i)} \cdot x) \quad (13)$$

The above-mentioned forces create a strong bias on the motion of the people. The people when following a robot also show behaviours that are seldom found and cannot be directly accounted for. As an example switching places, overtaking the robot, overtaking another person, lagging behind to spend more time at the older place, etc. To account for all such forces, a random force has been also utilized here (Eq. (14)),

$$\vec{f}_{rand}(h_{t,p}^{(i)}) = w_5 r \begin{bmatrix} \cos\theta_{rand} \\ \sin\theta_{rand} \end{bmatrix} \quad (14)$$

$$\vec{f}_{total}(h_{t,p}^{(i)}) = W_t \left(\vec{f}_{att}(R_t, h_{t,p}^{(i)}) + \vec{f}_{rep}(R_t, h_{t,p}^{(i)}) + \sum_{p'} \left(\vec{f}_{att}(q_{t,p'}, h_{t,p}^{(i)}) + \vec{f}_{rep}(q_{t,p'}, h_{t,p}^{(i)}) \right) \right) + (1 - W_t) \vec{f}_{rand} \quad (15)$$

When the human is being continuously detected, the weight has a high value favouring the attraction and repulsive forces as per modelling and is thus given by a normal distribution $W_t \sim N(W_{th}, \sigma_{th})$ with a large magnitude W_{th} and a small variance σ_{th} . However, once if the person is not being continuously detected, randomness needs to be added to indicate the increasing uncertainty in the system and is done by favouring the random forces by setting the weight as a uniform distribution as $W_t \sim U(W_{min}, W_{max})$ with relatively smaller values.

The force values are converted into the desired linear and angular speeds. The linear speed is given by the magnitude of the force, which is subjected to the constraints of the human walking speed, given by Eq. (16)

$$v(h_{t,p}^{(i)}) = \min \left(\left\| \vec{f}_{total}(h_{t,p}^{(i)}) \right\|, v_{max} \right) \quad (16)$$

The humans are holonomic and do not have kinematic constraints with respect to orientation. Therefore the humans are said to move in the desired angle modelled by the force, given by (Eq. (17))

$$\theta(h_{t,p}^{(i)}) = \text{atan2}(\vec{f}_{total}(h_{t,p}^{(i)}) \cdot y, \vec{f}_{total}(h_{t,p}^{(i)}) \cdot x) \quad (17)$$

The anticipated position of the human is (Eq. (18))

$$h_{t+1,p}^{(i)} = h_{t,p}^{(i)} + v(h_{t,p}^{(i)}) \delta_t \begin{bmatrix} \cos\theta(h_{t,p}^{(i)}) \\ \sin\theta(h_{t,p}^{(i)}) \end{bmatrix} \quad (18)$$

Here δ_t is the time period between the consecutive iterations and is the inverse of the tracking frequency.

Here $r \sim U[0,1]$ is a random magnitude of the force taken from a uniform distribution between 0 and 1; and $\theta_{rand} \sim U[-\pi, \pi]$ is a random direction of the force taken from a uniform distribution. w_5 is the relative importance of the force.

The attractive and the repulsive forces add up. However, the random force is uncertain. The force is seldom applied, but once applied has a strong effect. Therefore, a weighted fusion of the random force is made with all the other forces, where the weight is taken as $W_t \in [0,1]$. The different particles shall have a different weight of the fusion and therefore different particles shall represent different hypotheses, while the one that matches the observation shall eventually have a large weight. The total social force is given as Eq. (15)

4.2 Observation model

Our particle social force model anticipates the human position in the 3D real world. The role of the observation model is to weigh or evaluate every particle for its goodness. The weight is high if the particle agrees with the observation and poor otherwise. We have a rear looking camera and an observation model is applied whenever the camera detects the human. It is assumed that the robot’s pose is known and hence the pose of the camera is computable. The particle represents the position of the person. The position represented by the person is projected onto the known image plane of the camera. If the particle represents an accurate position of the person, the projected position on the image plane and the actual observation should closely match and vice versa. The error between the projected position of the person on the image and the actual observed position on the image is used to weigh the particle.

Let us study the projection of the actual human’s face into the 2D image plane as observed by a camera attached to the robot and facing at the rear. We have used a monocular camera for this model. Firstly, calibration is performed. A distinct image of the chessboard image has been captured with different view angles and distances for finding various camera intrinsic parameters and lens distortion. Hence standard calibration matrix (C) is given by Eq. (19) which is used in our camera model,

$$C = \begin{pmatrix} f_x & p_s & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (19)$$

Here (c_x, c_y) is the principal point offset and f_x and f_y represents the focal length while \mathcal{D}_s is axis skew (shear distortion in the projected image).

The robot coordinate frame is fixed to the centre of the differential wheel drive of the robot. The localization module available with the robot gives the transformation between the robot and the world coordinate axis system at any point of time. The camera is facing the rear and thus has a rotation of π along the Z-axis, and is at a height of h_c . This gives the transformation between the camera and the world coordinate axis as expressed as Eq. (20)

$$T_{camera}^W = \begin{pmatrix} \cos(R_t \cdot \theta + \pi) & -\sin(R_t \cdot \theta + \pi) & 0 & R_t \cdot X \\ \sin(R_t \cdot \theta + \pi) & \cos(R_t \cdot \theta + \pi) & 0 & R_t \cdot Y \\ 0 & 0 & 1 & -h_c \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (20)$$

Here $R_t \cdot \theta$ is the robot's orientation.

As per conventions, the Z-axis of the camera goes out of the camera and faces the human. Correction to the same conventions is expressed by the transformation given as Eq. (21),

$$T_{camera}^{camera'} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (21)$$

The human prospectively at position $h_{t,p}^{(i)}$ should hence be visualized at a position $T_{camera}^W h_{t,p}^{(i)}$ in the camera coordinate axis system which is further corrected to $T_{camera}^{camera'} T_{camera}^W h_{t,p}^{(i)}$ to correct the axis. Here projection onto image takes place by passing through the camera calibration matrix and producing the image point as to given in the Eq. (22–24)

$$\begin{bmatrix} u' \\ v' \\ s' \end{bmatrix} = C T_{camera}^{camera'} T_{camera}^W h_{t,p}^{(i)} \quad (22)$$

$$u_{t,p}^{(i)} = \frac{u'}{s'} \quad (23)$$

$$v_{t,p}^{(i)} = H - \frac{v'}{s'} \quad (24)$$

Here H is the height of the image and the subtraction is used to denote the change that the object detection uses the top-left as the origin, while the camera uses the bottom left as the origin.

The robot takes an image from the camera which passes through face detection and recognition module to identify the different people. Suppose the p^{th} person is observed in the image as $z_{t,p} = (u_{t,c(o)}, v_{t,c(o)})$, where c is the correspondence function. The weight of the particle is taken proportional to

the error in observation given by Eq. (25)

$$w_{t,p}^{(i)} = \eta \exp \left(\frac{\left(u_{t,c(o)} - u_{t,p}^{(i)} \right)^2 + \left(v_{t,c(o)} - v_{t,p}^{(i)} \right)^2}{2\sigma'^2} \right) \quad (25)$$

Here η is the normalization term set such that the sum of all weights is unity.

4.3 Detecting people leaving the group

The people in the group may often leave, which needs to be detected and the tracks need to be deleted. In this section, we examine this human behaviour and find out whether a human has left the group or not. The intuition is that a person not visible for a long time in the camera is assumed to have left the group. However, sometimes the invisibility would be because of contextual reasons. If the robot is making a sharp turn, it is evident that the person would not be able to run behind the robot. Similarly, if the person is following the robot, the person would be making a similar sharp turn after the robot. In such cases, the invisibility is contextual and such cases are filtered out from being deleted.

Let $visible(t)$ denote whether the human was detected at time t or not. $\neg visible(t)$ does not mean that the person has left the group, as the detection algorithm may have temporarily failed. If any human is not available in a group for a long duration of time then the human is said to have left the group at time t , given by Eq. (26).

$$left(t) = \neg visible(t') \forall t - \Delta \leq t' \leq t \quad (26)$$

Here Δ is the threshold of time within which if the person is not seen, the person is said to have left the group. A large value of the parameter reduces false positives; however, the system deletes the track a long time after the person has left the group. A small value of the parameter may state the person to have left the group even if the detector temporarily failed due to a sudden glare.

This condition is widely used in research. The condition will delete people when the robot is circumventing an obstacle or making a turn, during which time the visibility of the person is lost. The robotic journey also includes various turns and corners to achieve its goal while leading a group of humans. The workspace may have very less space to allow people to ideally follow the robot. When the robot is taking a turn to achieve its next sub-goal, even due to the limitation of space the human cannot stand exactly at the rear end of the robot and also cannot follow it ideally. In these circumstances the robot can be misled that the ‘‘human has left the group’’ or ‘‘not following it’’, however, the human is following a robot and has not left his group.

Thus the situations are detected when the robot is making a turn due to either of these reasons. Let the (G_r, X, G_r, Y) be the sub-goal that the robot is currently seeking. One expects the robot to always keep an angle of approximately, $atan2(G_r, Y - R_r, Y, G_r, X - R_r, X)$ to directly seek the goal under normal obstacle-free criteria. Some diversion is possible due to noises. However, if the current orientation of the robot is very different from this value, it means that the robot has made or is making a sharp turn for whatever reason possible.

$$turn(t) = true \text{ iff } \cos(\alpha_e(t)) < \tag{27}$$

$$\alpha_e = atan2(G_r, Y - R_r, Y, G_r, X - R_r, X) - R_r, \theta \tag{28}$$

Here the coordinates of the sub-goal that is being pursued by the robot are denoted by (G_r, X, G_r, Y) while the coordinates of the robot are denoted by (R_r, Y, R_r, Y) with the orientation of the robot as R_r, θ . δ is the maximum deviation in the natural navigation of the robot and a deviation more than this amount is said to denote a turn.

The leaving condition is therefore modified as Eq. (29)

$$left(t) = \neg visible(t') \wedge \neg turn(t') \forall t - \Delta \leq t' \leq t \tag{29}$$

It is important to note that after the robot has completed the turn, the people will still not be visible as they are now making the turn, following the footsteps of the robot. However, humans are fast in doing so and the cumulative relaxation of Δ gives enough time for humans to re-join the group.

4.4 Track re-initialization

Imagine a human left the group and is not seen by the camera. Naturally, the human would have left for any reason, however, since the human has now re-joined, the tracking must continue. Since the track was deleted when leaving the group, a new track is initialized based on the visible position of the human on the image. As the human re-joins the group, the track is re-initialized to start continuous tracking. Here the initial position of the human is to be determined from the 2D image plane, as the human is detected in the 2D image plane. The particle filter is initialized which needs to determine the position of the person in the 3D real world. An inversion of the observation model gives Eq. (30)

$$h_{t,p}^{(i)} = (T_{camera}^W)^{-1} (T_{camera}^{camera})^{-1} C^{-1} \begin{bmatrix} u' \\ v' \\ s' \end{bmatrix} \tag{30}$$

Knowing the height of the person directly gives the scale factor (s'). Knowing the position $(u_{t,p}^{(i)}, v_{t,p}^{(i)})$ where the person was observed in the image gives Eq. (31–32)

$$u' = s' u_{t,p}^{(i)} \tag{31}$$

$$v' = s' (H - v_{t,p}^{(i)}) \tag{32}$$

Plugging these values also gives the X and Y coordinates of the person.

5 Algorithm design

We have implemented a multi-behavioural social particle filter for 3D human tracking performed from a mobile robot while the human is also following the robot. Here our model initializes from the initial position of the humans and starts tracking. When any human leaves the group then the track is deleted and as the human re-joins its group, the track is re-initialized, and the tracking continues. The model is described in Fig. 2. The term ‘‘Social force based motion model’’ is used to refer to the computations described in Sec. 4.1. The model is used to compute the new position of a particle (hypothesis of the human) given the positions of the other persons and the robot. The previous position of the particle is known. The particle experiences attraction and repulsion forces from all other persons and robot. The particle is moved based on the same forces, that represents the new position of the particle.

If the person is detected in the image, the particles are re-weighted and re-sampled. The particles that as per the hypothesis project the person into the image closer to the observed location of the person are likely better and have a larger weight. This increases the particle density near the true position of the human. In case of an occlusion or a lack of detection for any other reason, no re-weighting is done. This means that the particles will keep moving as per the motion model, however, there will be no way to correct for a deviation that may arise along with time. Once the occlusion is removed, the re-weighting and re-sampling continue to correct the sample density around the true position of the person.

If a person is far off, he/she is unlikely to be detected by the face recognition library [41, 42]. Furthermore, a person far off will also not be visible in the camera. In such cases, the robot will not detect the person for a long time and the track may get deleted. Practically, the robot moves slower than the person. When the person comes closer and gets visible, a new track will be initialized.

The complete pseudo-code is given as Algorithm 1. The algorithm uses a timer to check the time period within which the person p was not visible. The flag *deleted* associated with every person checks if the person is within the group and being tracked, or its track is deleted for leaving the group. $q_{t,p}$ is the tracked position of the person p at time t , which is used to compute the attraction and repulsion forces between the people.

Algorithm 1: Tracking

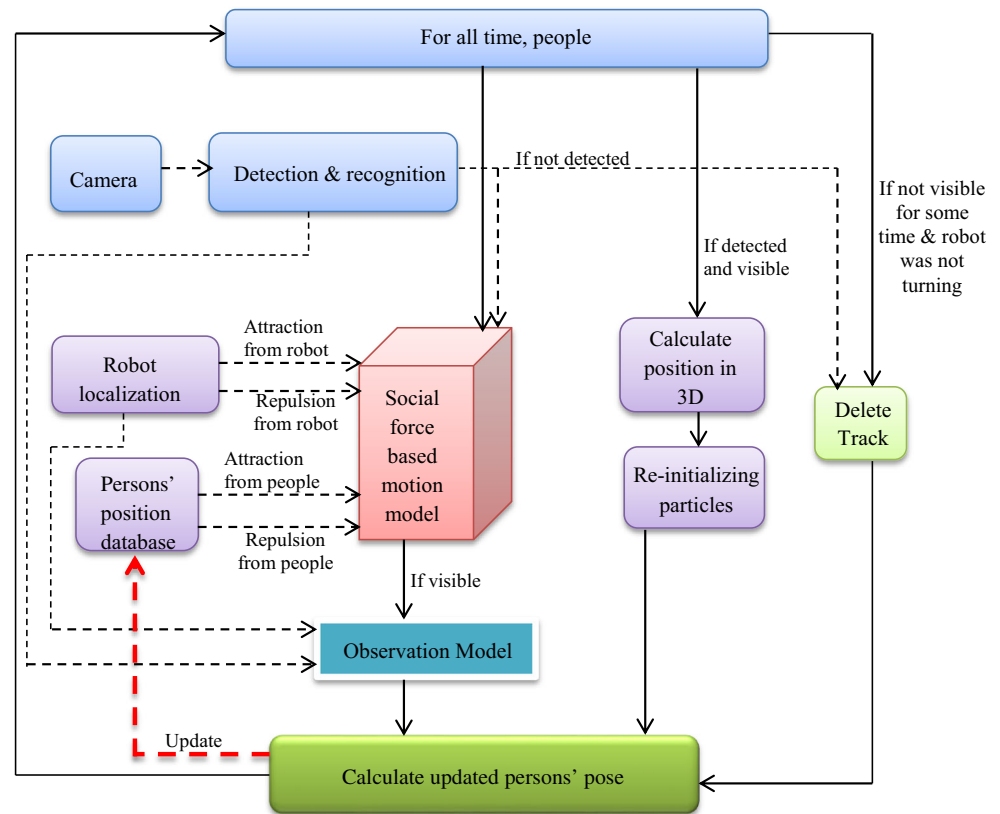
Algorithm 1 demonstrates the entire procedure of the 3D-PSF model for multi-humans, trajectory prediction and tracking in a real-time. Initially, every human is

1. For all persons p in group
2. Deleted(p) \leftarrow false
3. Start NotVisibleTimer(p)
4. End For
5. For all persons p detected at $(u_{0,p}, v_{0,p})$ and recognized in the first frame
 6. Calculate position of person $q_{0,p}$ using (30-32), or

$$q_{0,p} = (T_{camera'}^W)^{-1} (T_{camera'})^{-1} C^{-1} \begin{bmatrix} u' \\ v' \\ s' \end{bmatrix}, u' = s' u_{0,p}, \text{ and } v' = s' (H - v_{0,p})$$
 7. Initialize particles $h_{t,p}^{(i)} = q_{0,p} + noise, w_{t,p}^{(i)} = 1/n$, where n is the number of particles
 8. Reset NotVisibleTimer(p)
9. End For
10. While (tracking)
 11. For all people p being tracked
 12. Get visible(p), $u_{t,p}, v_{t,p}$ (whether p is visible, and if yes the position in the image) from the face detection and recognition library
 13. If Deleted(p) \wedge Visible(p)
 14. Calculate position of person $q_{t,p}$ using (30-32), or

$$q_{t,p} = (T_{camera'}^W)^{-1} (T_{camera'})^{-1} C^{-1} \begin{bmatrix} u' \\ v' \\ s' \end{bmatrix}, u' = s' u_{t,p} \text{ and } v' = s' (H - v_{t,p})$$
 15. Initialize particles $h_{t,p}^{(i)} = q_{t,p} + noise, w_{t,p}^{(i)} = 1/n$, where n is the number of particles
 16. Reset NotVisibleTimer(p)
 17. Deleted(p) \leftarrow false
 18. End If
 19. If \neg Deleted(p)
 20. For all particles $h_{t,p}^{(i)}$,
 21. Calculate $\overrightarrow{f_{attr}}(\mathbf{R}_t, \mathbf{h}_{t,p}^{(i)})$ and $\overrightarrow{f_{rep}}(\mathbf{R}_t, \mathbf{h}_{t,p}^{(i)})$ using (6-9)
 22. Calculate $\overrightarrow{f_{attr}}(\mathbf{q}_{t,p}, \mathbf{h}_{t,p}^{(i)})$ and $\overrightarrow{f_{rep}}(\mathbf{q}_{t,p}, \mathbf{h}_{t,p}^{(i)})$ using (10-13)
 23. Calculate $\overrightarrow{f_{rand}}(\mathbf{h}_{t,p}^{(i)})$ using (14)
 24. Calculate $\overrightarrow{f_{total}}(\mathbf{h}_{t,p}^{(i)})$ using (15)
 25. Calculate $h_{t+1,p}^{(i)}$ using (16-18)
 26. End For
 27. End If
 28. If \neg visible(p), Update NotVisibleTimer(p)
 29. Calculate Turn(t) using $turn(t) = true \text{ iff } \cos(\alpha_e(t)) < \delta$ and $\alpha_e = atan2(G_t.Y - R_t.Y, G_t.X - R_t.X) - R_t.\theta$
 30. If Turn(t) calculated using (27-28), Reset NotVisibleTimer(p)
 31. If NotVisibleTimer(p) $> \Delta$, Deleted(p) \leftarrow true (algorithmic implementation of (29))
 32. If visible(p)
 33. Calculate $u_{t,p}^{(i)}$ and $v_{t,p}^{(i)}$ using (19-24)
 34. Calculate $w_{t+1,p}^{(i)}$ using (25)
 35. Resample $h_{t+1,p}^{(i)}, w_{t+1,p}^{(i)}$
 36. End If
 37. If \neg deleted(p), $q_{t,p} = \sum_{i=1}^n w_{t,p}^{(i)} h_{t,p}^{(i)}$
 38. Else $q_{t,p} = null$
 39. End If
 40. End For
 41. End While

Fig. 2 Design of multi-behavioural 3D-PSFM



detected by their faces in the 2D image plane and our proposed model converts the initial position of every human in the 3D real-world coordinates in lines 5–7. Lines 1–3 start timers useful for the deletion function, however, line 8 resets the timers for the persons visible initially. If a deleted person is further detected in the 2D image plane, then their 3D position is computed from the inverse camera transformation method in lines 13–17. This 3D position is used to re-initialize the tracker for again performing tracking and prediction after track's deletion. If a person is not deleted, lines 20–26 apply the motion model by using the social force method. Lines 28–31 handle the condition when a person should be deleted because of a lack of visibility for consecutively Δ times. Lines 29–30 handle the special case when the robot is making a turn, while line 31 implements the deletion logic. As persons are detected in the subsequent frame then their position in the 2D image plane is determined using 3D real world to 2D image plane transformation (conversion) using lines 31–35. Line 34 uses the visibility information to update the weights while line 35 resamples using the updated weights. Lines 37–40 compute the most probable person location based on the particle's position.

6 Experiment evaluation and result analysis

The proposed work is tested on real-life settings at the Centre of Intelligent Robotics, IIT Allahabad with the AmigoBot robot and a rear-facing webcam, where a few lab members were asked to casually follow the robot, while behaving erratically at times. The Amigobot robot was fixed at a pre-determined source. The camera was mounted on top of the robot. AmigoBot robot was used to track humans. AmigoBot robot was stationed at the origin (0,0) and the humans stand behind the robot with a certain distance. The coordinates of the humans are taken as the (x, y) position of the humans and the height of humans is replaced as the z coordinate. The transformation between the robot base and the camera was manually measured. Camera calibration was done that helps to convert the 3D point (real-plane) into the 2D point (image-plane). Thereafter, different processes run nearly independently on the same system. The first process is the teleoperation of the robot. The teleoperation is done by using robot-specific libraries. As the robot moves, a service provided by the vendor keeps computing the position of the robot. The position is computed by using the high-precision encoders on the robot. Finally, a separate module runs the particle filter proposed in the paper. The development is done using the Python

Table 1 Parameters values

SN	Parameters	Value	Remarks
1.	Number of particles	200	Set by trial and error
2.	Robot radius	0.2533 m	As per actual measurement
3.	Person radius	0.2533 m	As per actual measurement
4.	Average velocity of the human	1.500 m/s	As per actual measurement
5.	Height of the camera	1.220 m	As per actual measurement
6.	Calibration matrix	$\begin{pmatrix} 718.46 & 0.0 & 314.62 & 0.0 \\ 0.0 & 719.49 & 265.41 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{pmatrix}$	Obtained from calibration
7.	Image resolution	(640×480)	As per actual
8.	Coefficient of person to robot attraction force (w_1)	1.3181	Set so that the attraction force is at a set threshold when the person is at a pre-determined social distance from the robot
9.	Coefficient of person to robot repulsion force (w_2)	2473.082	Set so that attraction and repulsion forces cancel each other when the person is at a pre-determined social distance from the robot
10.	Coefficient of person to person attraction force (w_3)	2.0251	Set so that the attraction force to the person is 1/3rd of the attraction force to the robot when the person is at a pre-determined social distance from the robot and another person
11.	Coefficient of person to person repulsion force (w_4)	824.361	Set so that attraction and repulsion forces cancel each other when two persons are at a pre-determined social distance from each other
12.	Coefficient of random force (w_5)	1500.0	Set to make the force in the same range as the other forces
13.	Coefficients related to the total force	$W_{th}=0.9, W_{min}=0, W_{max}=0.2;$	As per program logic
14.	Minimum angle to qualify a turn (δ), as a Cosine value	0.6	Set by measuring the angle that the robot makes at different turns in the environment
15.	Minimum time of invisibility to delete a track (Δ)	4 s.	Set by measuring the actual time that the humans left in the experiments due to invisibility due to genuine reasons apart from corners

programming language. The camera continuously takes images. The humans are detected by their faces using a convolution neural network [21, 22]. The initial position of the particles is computed using an inverse transform of image coordinates to a person's position for each person. Thereafter, the

program runs the Particle Filter logic inside a loop. At every iteration, the camera takes a new image as input and the face detection is done. The particles are moved based on the position of the robot computed from the encoders and the published positions of the different persons. If the person is

Table 2 Comparisons of the proposed approach

Scenario ID	Person ID	Proposed (mm)	Random Particle filter (mm)	Tracking in Image (mm)	Kalman Filter (mm)	LSTM (with re-initialization) (mm)	LSTM (without re-initialization) (mm)
I	1	629.28	662.26	4135.63	998.69	1950.71	2249.56
	2	618.69	670.93	4410.85	985.54	3147.95	3448.09
	3	258.24	601.85	5128.56	958.67	3400.55	3682.45
II	1	234.88	303.87	10,577.32	802.64	4802.54	5604.49
	2	229.96	284.24	1767.64	706.27	8638.65	9129.27
	3	388.56	468.88	2894.84	1508.49	8612.41	8932.18
III	1	406.27	738.69	7212.76	959.64	3126.22	6702.37
	2	676.32	739.63	30,610.39	1202.34	5465.19	8312.73
	3	385.18	810.94	6215.89	1284.75	9265.74	7798.63

Best results are shown in bold

Table 3 F-score for detection of human leaving the group

Scenario ID	Person ID	Proposed (with corner detection)	Proposed (without corner detection)
I	1	0.98	0.07
	2	0.98	0.46
	3	0.94	0.09
II	1	0.96	0.81
	2	0.98	0.78
	3	0.96	0.79
III	1	0.97	0.68
	2	0.98	0.55
	3	0.98	0.61

Best results are shown in bold

Table 4 Confusion matrix

Scenario ID	Person ID	Proposed (with corner detection)				Proposed (without corner detection)			
		TP	TN	FP	FN	TP	TN	FP	FN
I	1	98	10,827	0	42	90	8777	2020	80
	2	1026	9911	0	30	1079	7392	2425	71
	3	960	9897	77	33	125	8589	2174	79
II	1	3138	12,541	0	200	3128	11,364	1177	210
	2	2247	13,552	0	80	2245	12,389	1163	82
	3	2967	12,712	0	200	3018	11,303	1409	149
III	1	2500	22,659	2	120	2495	20,466	2193	125
	2	2283	22,916	2	80	2285	19,388	3528	78
	3	2142	23,059	0	80	2135	20,477	2580	87

Best results are shown in bold

detected, the observation model is applied. The deletion of track and the re-initialization happens by handling the different timers set for the purpose as per the logic. Every human in a group usually follow the robot and due to their socialistic behaviour when any human leaves its group, then the track is

deleted for that human. In the continuation of the navigation process as the human join their group, the human is detected in the 2D image plane. This 2D image plane coordinate is converted into 3D real-plane using inverse camera calibration and inverse transformation. This 3D coordinate is the re-initialized

Table 5 Probabilistic interpretation

Scenario ID	Person ID	Proposed (with corner detection)			Proposed (without corner detection)		
		Precision	Recall	Accuracy	Precision	Recall	Accuracy
I	1	1.00	0.70	0.99	0.04	0.30	0.05
	2	1.00	0.97	0.99	0.52	0.93	0.61
	3	0.92	0.96	0.98	0.80	0.77	0.79
II	1	1.00	0.94	0.98	0.72	0.93	0.91
	2	1.00	0.96	0.99	0.65	0.96	0.92
	3	1.00	0.93	0.98	0.68	0.95	0.90
III	1	0.99	0.95	0.99	0.53	0.95	0.90
	2	0.99	0.96	0.99	0.39	0.96	0.85
	3	1.00	0.96	0.99	0.45	0.93	0.89

Best results are shown in bold

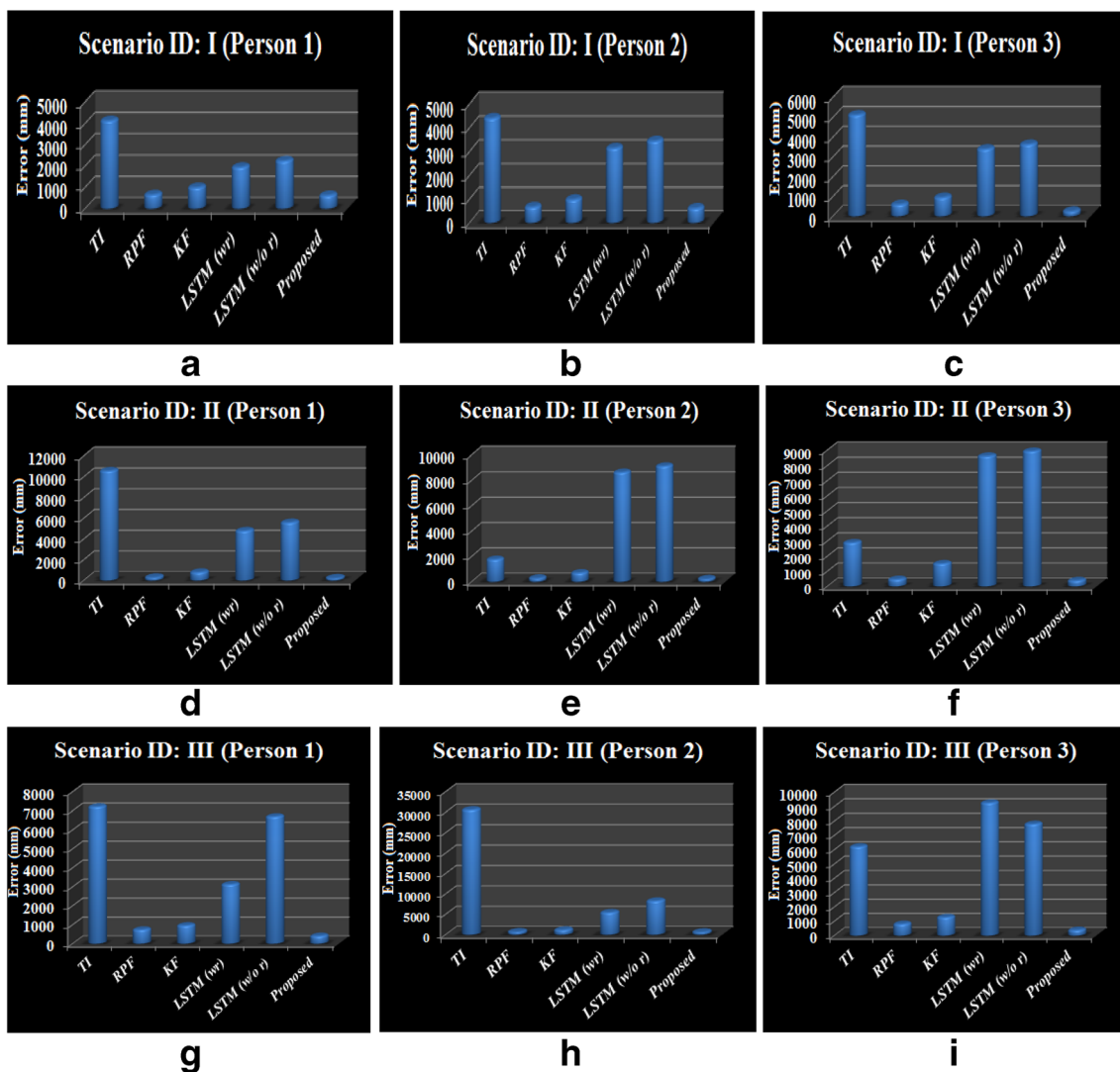


Fig. 3 Root mean square error (TI: Tracking in Image, RPF: Random Particle Filter, KF: Kalman Filter; LSTM (wr): LSTM (with re-initialization), LSTM (w/o r): LSTM (without re-initialization)) (a) Scenario ID: I Person 1 (b) Scenario ID: I Person 2 (c) Scenario ID: I

Person 3 (d) Scenario ID: I Person 1 (e) Scenario ID: II Person 2 (f) Scenario ID: II Person 3 (g) Scenario ID: III Person 1 (h) Scenario ID: III Person 2 (i) Scenario ID: III Person 3

point of the human and re-assigned to the tracker for further tracking. After all computations, the weighted addition of all particles is used to compute the position of the person, and the same is logged and published.

The real-life experiments do not have the ground truth of the tracked persons. Therefore, the experiments were also done on simulated robots where one robot enacted the real physical robot with a simulated rear-facing web-camera, and

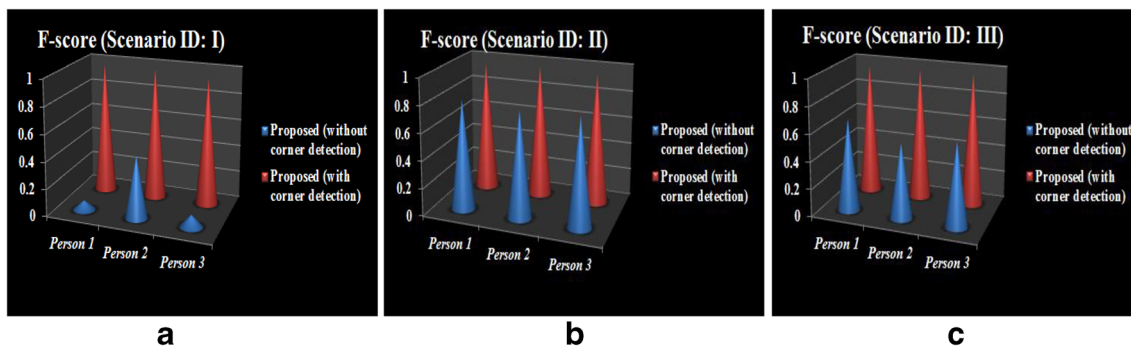


Fig. 4 F-score (a) Scenario ID: I (b) Scenario ID: II (c) Scenario ID: III

the other robots enacted by the following people. These robots were teleoperated and the teleoperating humans were asked to maintain as many social norms as possible. To make the

simulations realistic, the map used was that of the same place in which the socialistic experiments were done. The 2D map was made using a SLAM library using the Pioneer LX robot.

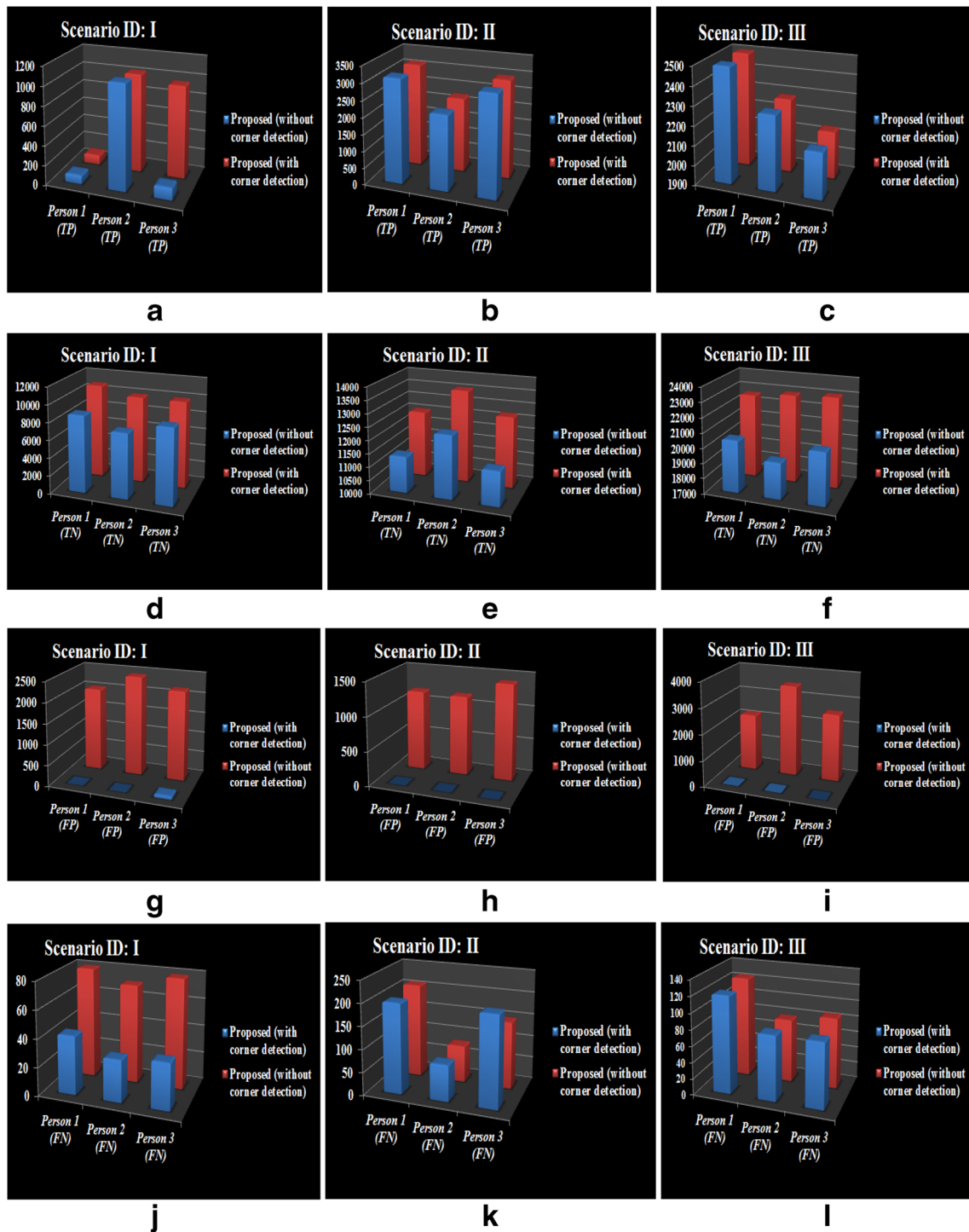


Fig. 5 Confusion matrix elements (a) Scenario ID: I True positives (b) Scenario ID: II True positives (c) Scenario ID: III True positives (d) Scenario ID: I True negatives (e) Scenario ID: II True negatives (f) Scenario ID: III True negatives (g) Scenario ID: I False positives (h) Scenario ID: II False positives (i) Scenario ID: III False positives (j) Scenario ID: I False negatives (k) Scenario ID: II False negatives (l) Scenario ID: III False negatives

The 3rd dimension was added by assuming the obstacles have a fixed common height. The simulated camera was restricted to the same constraints as the real camera, using the same projection matrix. The camera was programmed so as not to see through the obstacles. The camera was noised to imitate the large errors, including errors in detection and recognition, associated with the real camera. We used the Robot Operating System (ROS) library as a generic framework and MobileSim simulator for the Amigobots. First, the environment map was loaded on the MobileSim simulated. Four robots were placed in the MobileSim simulator. RosAria library was utilized which provides the basic functionalities of the simulated robots including integration with ROS. All the four robots have their coordinates, and one robot was made to behave like a leader robot and the remaining three robots acted like humans. Robots were traversed by teleoperation. Here four different

computing systems were needed to teleoperate the four robots (1 imitating the leader robot and 3 imitating the humans). One system was made as a ROS master system (that imitated the leader robot) and the remaining three systems connected to the ROS master system. We know the actual trajectory of all robots from the simulator and this trajectory was used as the ground truth for comparing our motion model. The trajectory was continuously logged. The same program of the particle filter was used, except that the camera image was taken by the simulated camera.

We first discuss the comparisons with the simulated setup and then discuss the experiments with the real-life humans under socialistic settings. All the behaviours are implemented in several scenarios and further our model is employed onto those different scenarios. Here three scenarios are presented which consist of all different commonly recurring human

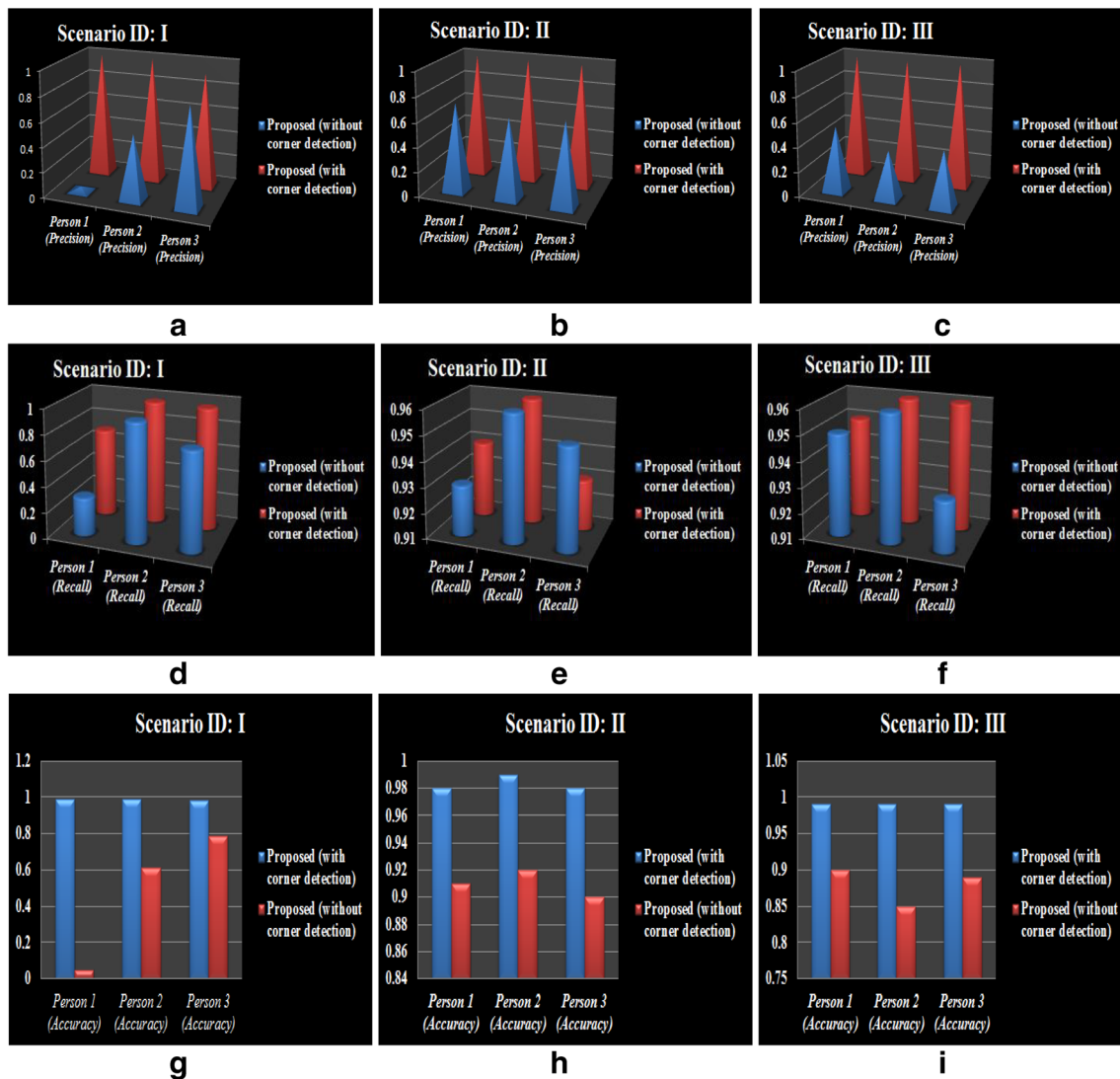


Fig. 6 Interpretation graphs (a) Scenario ID I Precision (b) Scenario ID II Precision (c) Scenario ID III Precision (d) Scenario ID I Recall (e) Scenario ID II Recall (f) Scenario ID III Recall (g) Scenario ID I Accuracy (h) Scenario ID II Accuracy (i) Scenario ID III Accuracy

behaviours such humans swapping, humans overtaking each other, human leaving the group, adjustment of humans accordingly due to limited availability of the space at corridors and corners, humans are too busy at the previous site, etc. The

general behaviour is the humans following the robot for their navigation. The robot knows only the initial position of the humans and anticipates the humans' trajectory in both the 3D real world and the 2D image plane.

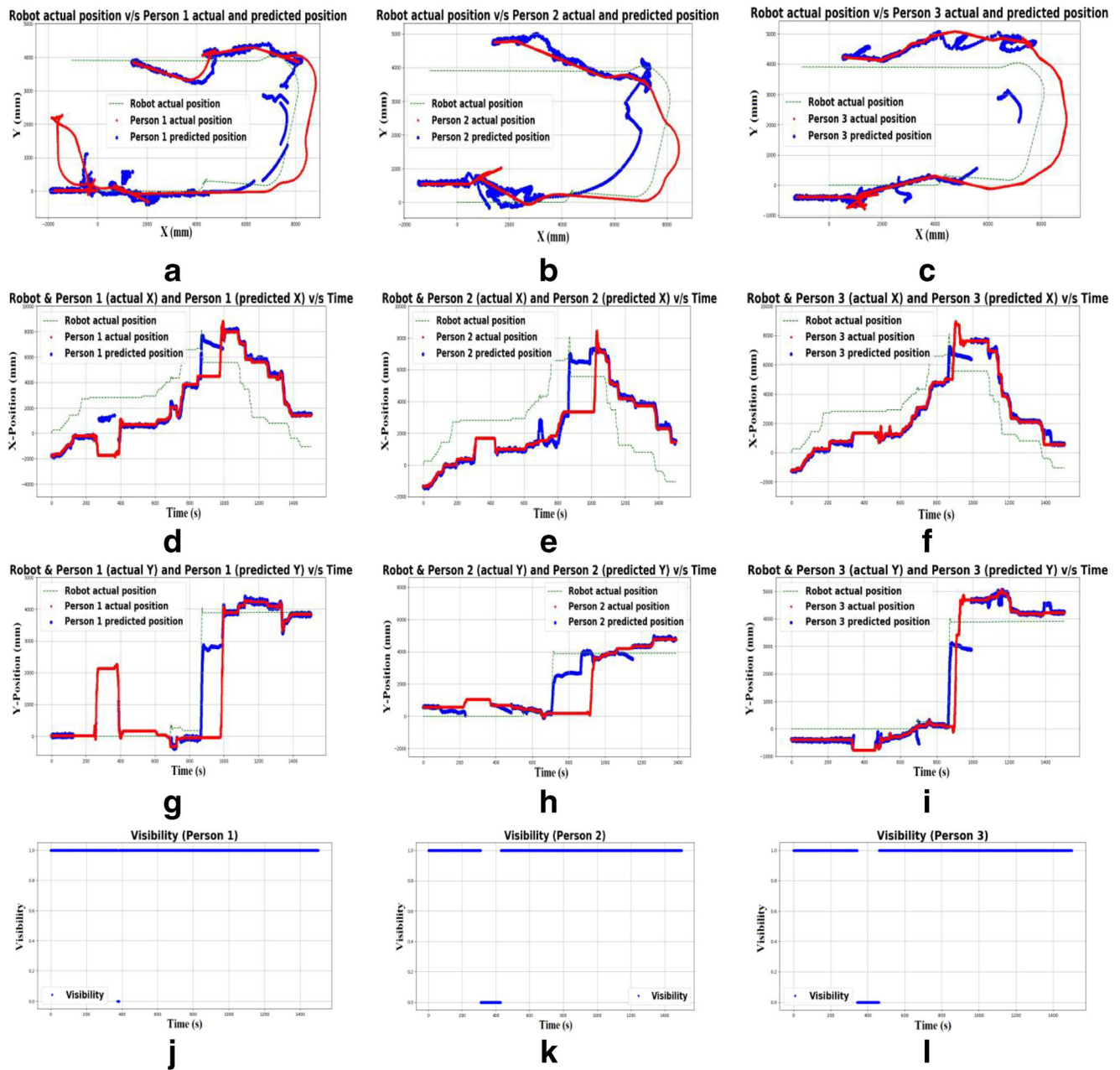


Fig. 7 Scenario ID: I 3D trajectory anticipation on a different plane using a Multi-behavioural 3D Particle Social Force Model (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (d) Robot trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s human

3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane (j) Visibility graph of human 1 (k) Visibility graph of human 2 (l) Visibility graph of human 3

One of the important aspects is the parameter setting. Some parameters were directly measurable like the radial occupancy of the robot and the person, the average walking speed of the humans, and the camera calibration matrix. There are numerous coefficients related to the forces and trial and error between all the coefficients could have been difficult. Hence, we first measured the distance that a human prefers to keep with the robots by doing some social experiments. At this social distance value, it is known that the attraction and repulsion forces between the person and the robot should cancel each other. We arbitrarily fix the attraction force coefficient value at this distance and set the repulsion force coefficient value to cancel the attraction force. Similarly, we also measured the social distance that two people tend to keep between themselves. Similar equilibrium conditions were considered between two people and the robot. Heuristically, we set the attraction between two people to be 1/3rd in contrast to the attraction between a robot and a person. This is based on the observation that while following, the people are primarily attracted towards the leader with a small attraction towards each other. The repulsion force coefficient is set such that the repulsion force cancels the attraction force between people at the social distance. The random force is scaled to the same scale. We performed several experiments to set the number of particles parameter. In general, a higher number of particles is preferable, which however also increases the

computation time. The parameter values are described in Table 1.

6.1 Performance evaluation and its comparison

To benchmark the dataset, we use 4 baseline approaches. The first baseline approach is a particle filter with a random motion model. The filter deletes track when the person is non-observable for a long time and re-initializes the track when the person is visible again. The second approach directly tracks the subjects into the image. However, knowing the height of the person, the conversion of the position from the 2D image to a 3D position in real-world is additionally done for comparisons. The next method is the Kalman Filter.

The other competent methods are from the trajectory prediction approaches. The LSTM based implementations are increasingly getting popular. In our experiments, the assumption is that the humans are only partially observable when they are within the field of view of the camera, and hence the training data is highly limited that does not cover most of the interesting cases when the human is outside the field of view. Even if it is critiqued that external sensors can be used to make training data, in our assumption the humans have unclear intents and stochasticity in their behaviour. Hence for the same situation, the person could react differently. Moreover, the persons occasionally leave and re-join groups which is a displayed behaviour but should not be learnt. Initially, an

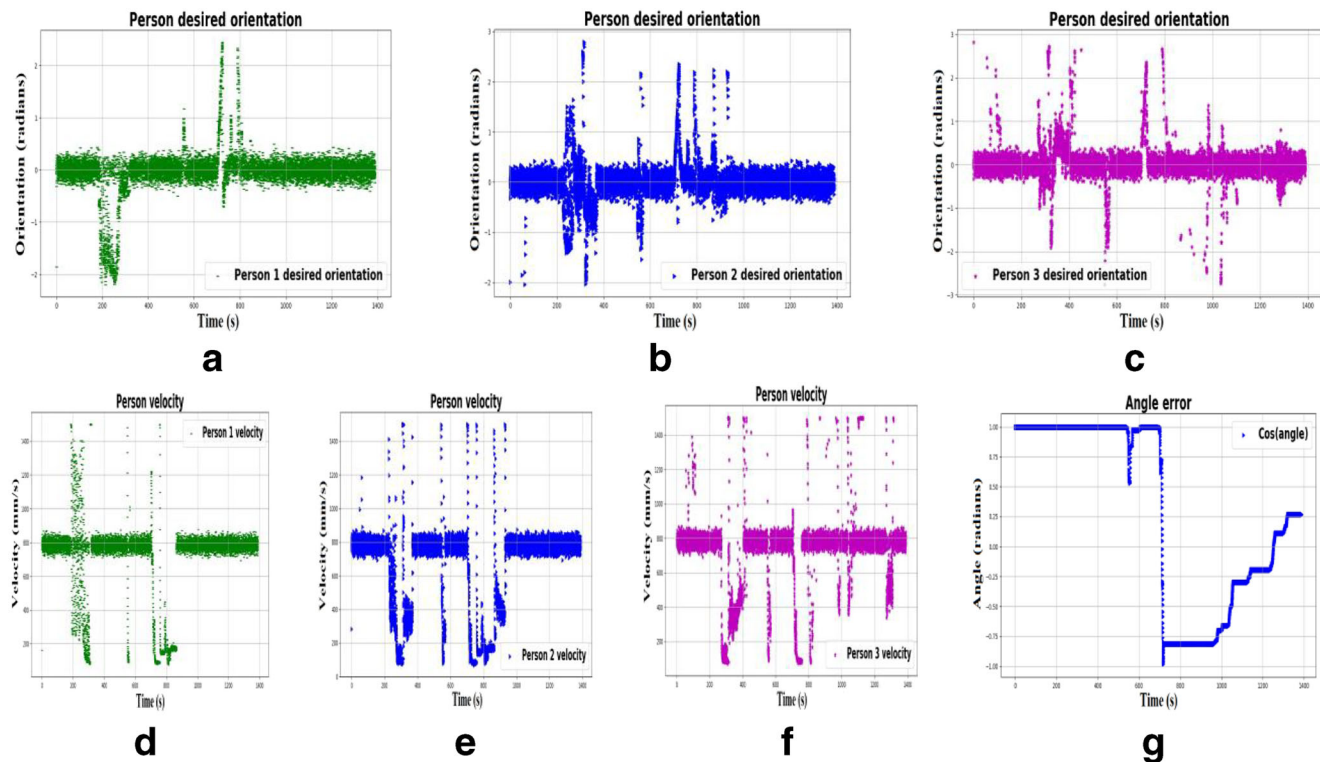


Fig. 8 Scenario ID: I Humans constraints (a) First human desired orientation (b) Second human desired orientation (c) Third human desired orientation (d) Velocity of first person (e) Velocity of second person (f) Velocity of third person (g) Corner angle error

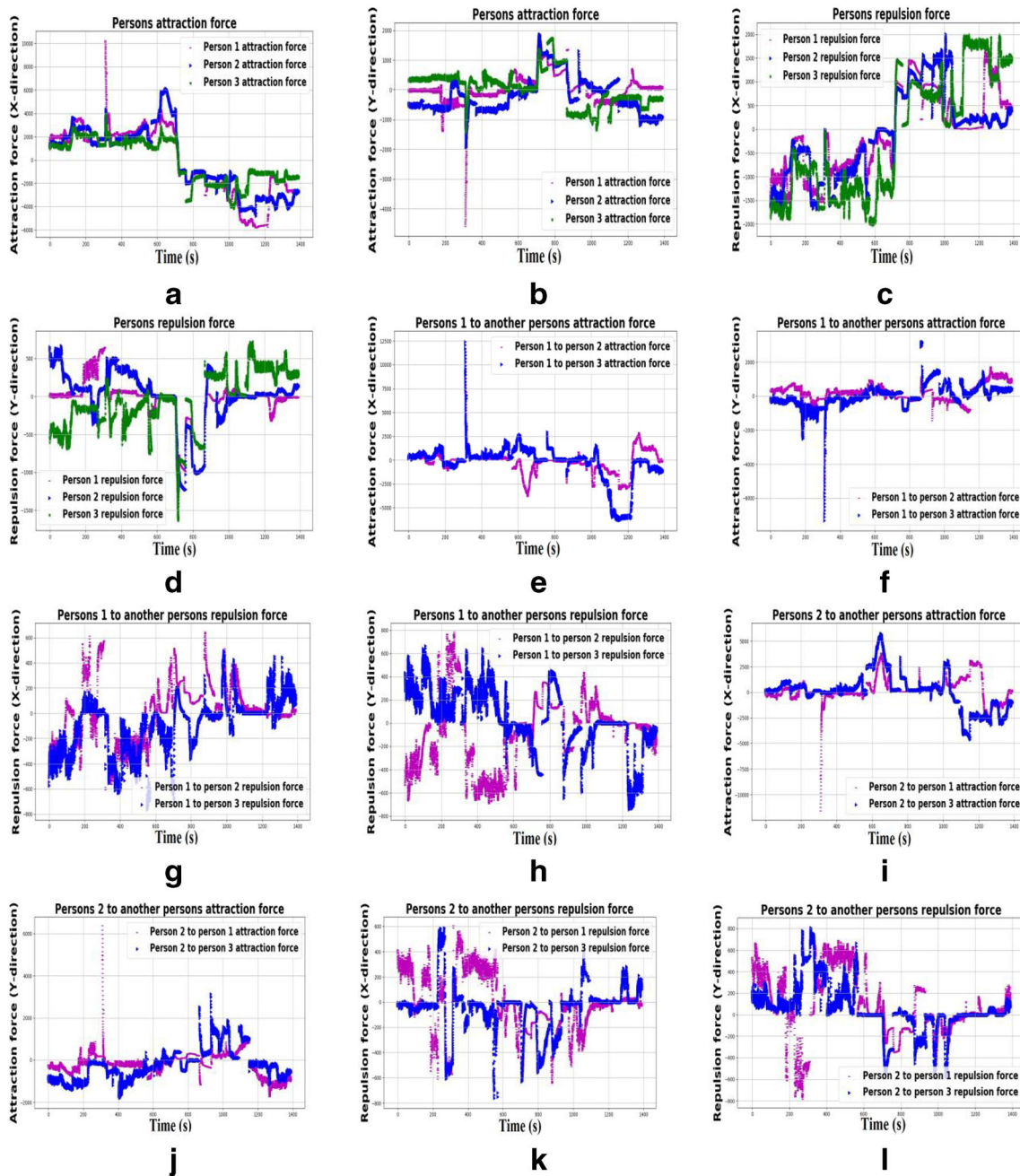


Fig. 9 Scenario ID: I Social forces modelling (a) Humans' attraction force towards the robot in the X-direction (b) Humans' attraction force towards the robot in the Y-direction (c) Humans' repulsive force towards the robot in the X-direction (d) Humans' repulsive force towards the robot in the Y-direction (e) Person 1 to another person's attraction force in the X-direction (f) Person 1 to another person's attraction force in the Y-direction (g) Person 1 to another person's repulsive force in the X-direction (h) Person 1 to another person's repulsive force in the Y-direction (i) Person 2 to another person's attraction force in the X-direction (j) Person 2 to another person's attraction force in the Y-direction (k) Person 2 to another person's repulsive force in the X-direction (l) Person 2 to another

person's repulsive force in the Y-direction (m) Person 3 to another person's attraction force in the X-direction (n) Person 3 to another person's attraction force in the Y-direction (o) Person 3 to another person's repulsive force in the X-direction (p) Person 3 to another person's repulsive force in the Y-direction (q) Random force on X-plane (r) Random force on Y-plane (s) Total social force applied on human 1 in the X-direction (t) Total social force applied on human 1 in the Y-direction (u) Total social force applied on human 2 in the X-direction (v) Total social force applied on human 2 in the Y-direction (w) Total social force applied on human 3 in the X-direction (x) Total social force applied on human 3 in the Y-direction

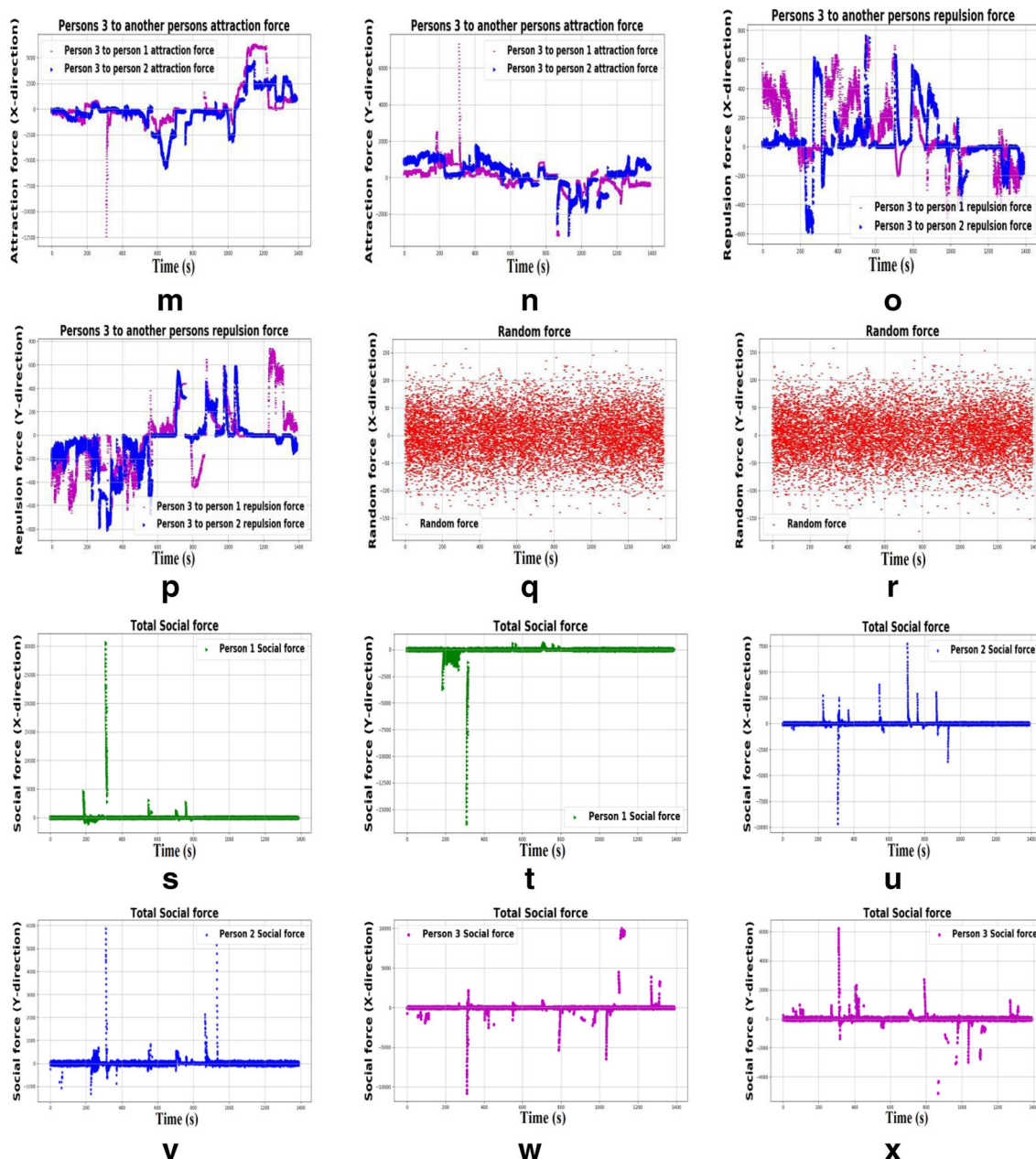


Fig. 9 (continued)

attempt was made to see if the LSTM can reject such noises and learn using simulated data. However, training on one person's data performed poorly when tested on the data of another person. This also highlights a known observation that different people behave differently. Hence, training was repeated when training and testing on the same person's data. The LSTM took the positions of the robots and all persons to predict the motion of the person under study.

An unfair disadvantage of the LSTM was that the other methods benefited from the continuous observations from the camera, while the LSTM only used the initial conditions along with the known position of the robot. Hence another

method was used where the LSTM was actively re-initialized from the position calculated from the image. The LSTM was used to predict a short sequence. Thereafter, the LSTM was fed a small trajectory computed from the detections in the image, and the LSTM was made to predict another short sequence.

Table 2 gives the error calculated for the humans' actual and tracked trajectory in 3D. The metric of comparison is the root mean squared error (RMSE) between the actual and the tracked positions of the humans. Each algorithm has an option of denial of service where it deletes the track assuming that the person has left. While calculating the RMSE, a penalty of

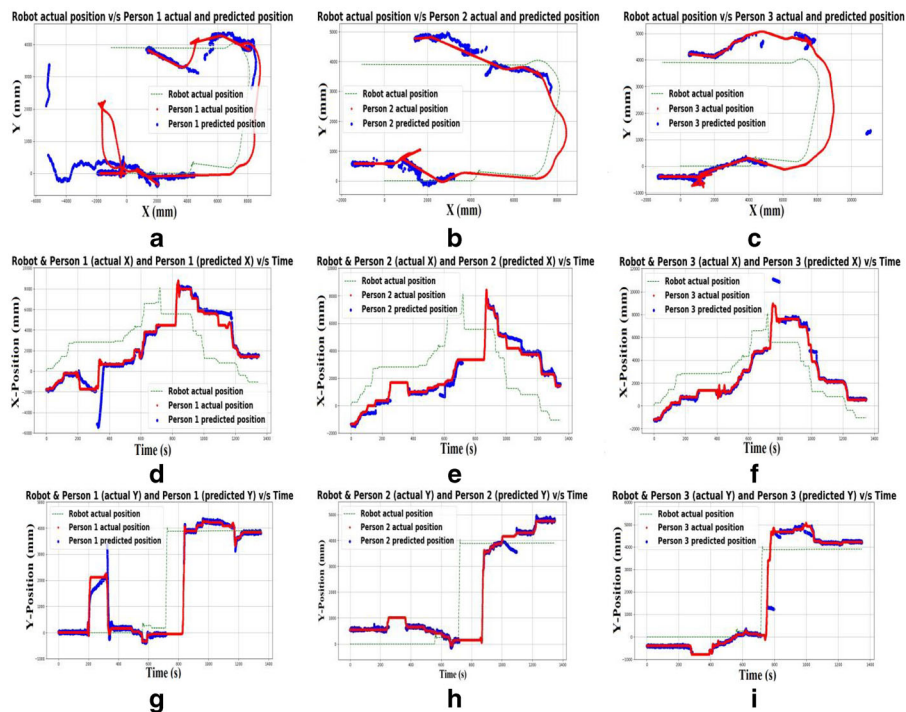


Fig. 10 Scenario ID: I 3D trajectory anticipation on a different plane using a Random particle filter (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (e) Robot trajectory v/s human 2 actual v/s

anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

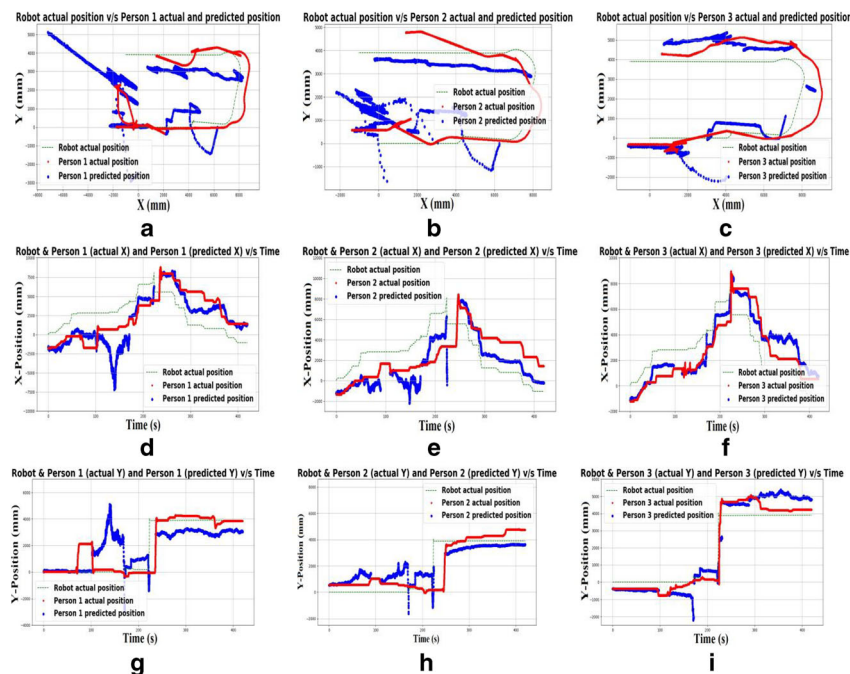


Fig. 11 Scenario ID: I 3D trajectory anticipation on a different plane using Tracking in image (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (e) Robot

trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

1400 is added for all cases when the person did not leave while the algorithm predicted so. Otherwise, the algorithms can decline service for the whole motion to get the least errors. We have done a statistical analysis of all the experiments reported. Pairwise T-test was used to understand the statistical significance of the proposed approach with all the other approaches. Based on statistical testing it was observed that the proposed algorithm is better than all other methods used for comparisons with a significance level of at least 95%.

We have tested our proposed method on three different scenarios to ensure the validity of our 3D particle model. In the comparison with other methods, our model gives very less error between 0.2 m to 0.6 m and other methods give a much larger error. Random particle filter method cannot be applied everywhere because it gives a large error for Scenarios I and III, while the errors are lower in Scenario II, still higher than the proposed approach. The performance of the random particle

filter deteriorates when the person is outside the visibility, while the filter soon erroneously deletes the track while the robot may have been combating a corner. When tracking in the image, the errors are extremely high because of the moving camera. The errors become very large as soon as the person is outside the visibility of the robot, in which case the position is guessed under the assumption of smoothness in the image, which is void considering that the robot was actually making a sharp turn and no motion of the human shall also result in a sharp motion in the image, which the filter erroneously smoothens out. The Kalman Filter performs poorly due to a linearity assumption that does not hold good in this model. As the robot rotates, the non-linearity significantly increases that invalidates the model. It must be stressed that the motion model in our approach is based on heuristics that act stochastically and hence there is no easy way to get the derivative for the implementation of the Extended Kalman Filter.

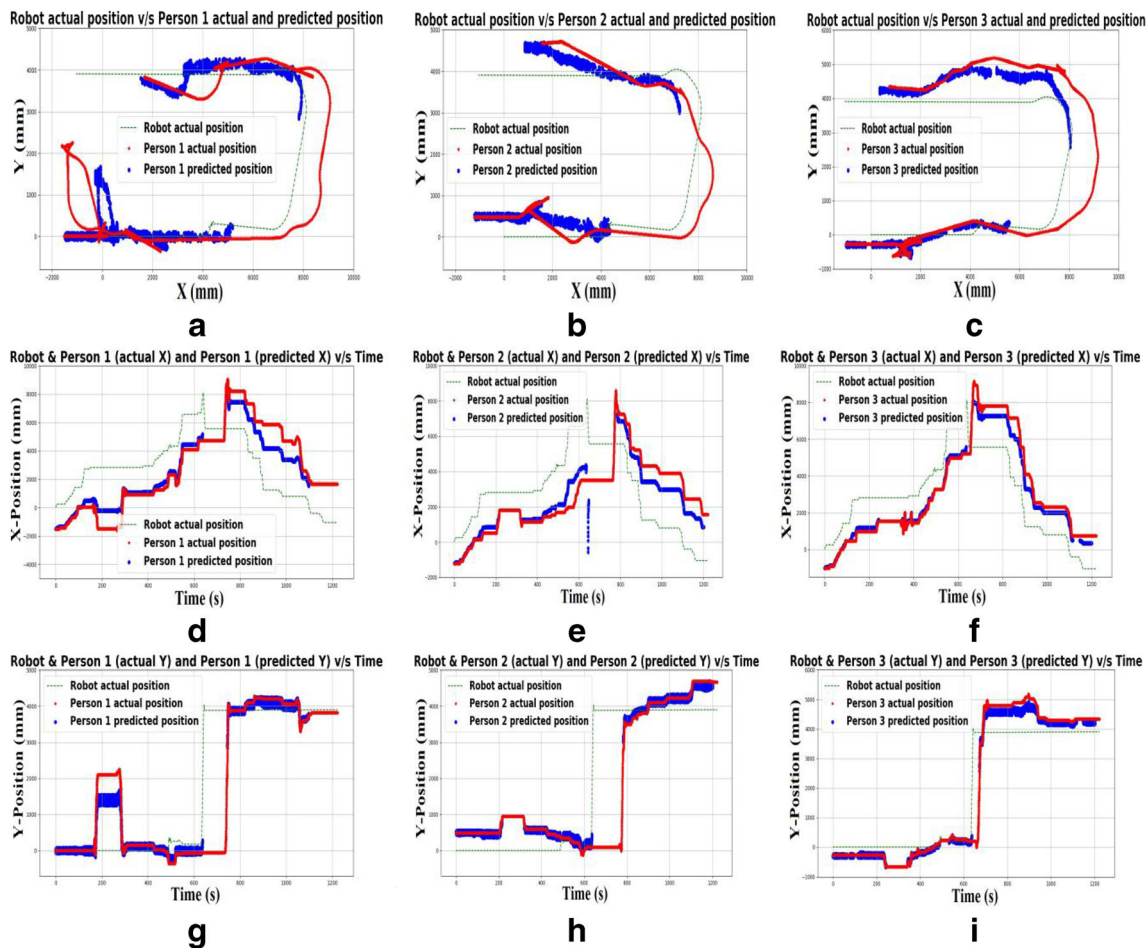


Fig. 12 Scenario ID: I 3D trajectory anticipation on a different plane using a Kalman Filter **(a)** Robot trajectory v/s human 1 actual v/s anticipated trajectory **(b)** Robot trajectory v/s human 2 actual v/s anticipated trajectory **(c)** Robot trajectory v/s human 3 actual v/s anticipated trajectory **(d)** Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane **(d)** Robot

trajectory v/s human 2 actual v/s anticipated trajectory on X-plane **(f)** Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane **(g)** Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane **(h)** Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane **(i)** Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

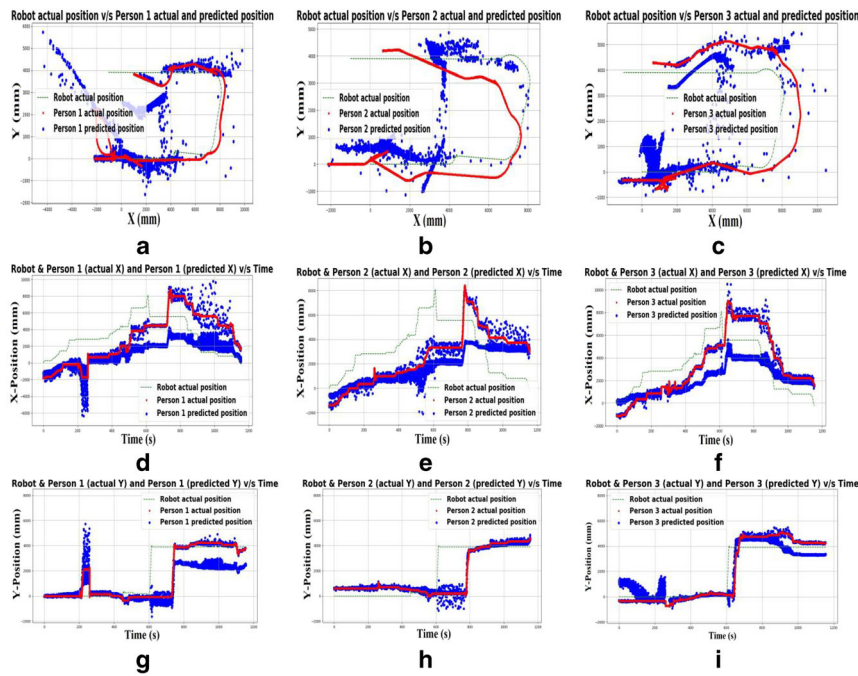


Fig. 13 Scenario ID: I 3D trajectory anticipation on a different plane using a LSTM (with re-initialization) (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (d)

Robot trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

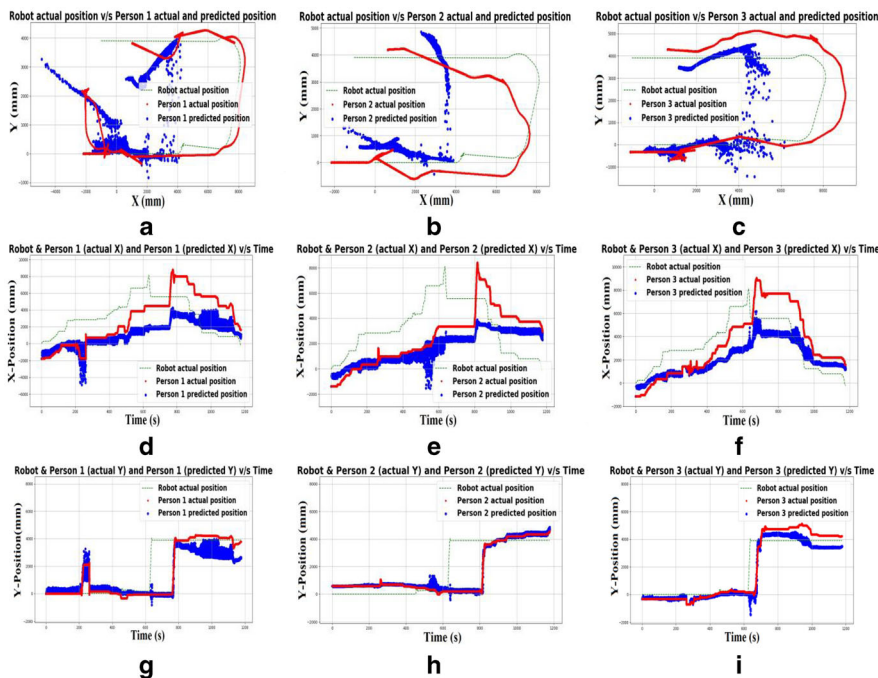


Fig. 14 Scenario ID: I 3D trajectory anticipation on a different plane using a LSTM (without re-initialization) (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (d) Robot trajectory v/s human 2 actual v/s

anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

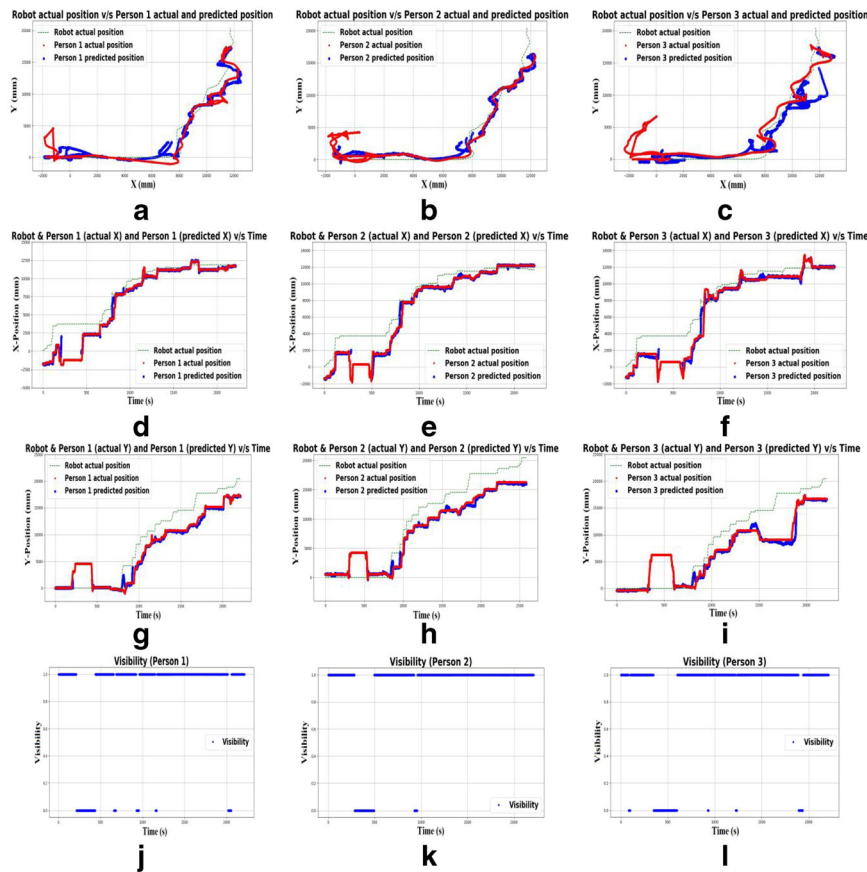


Fig. 15 Scenario ID: II 3D trajectory anticipation on a different plane using a Multi-behavioural 3D Particle Social Force Model (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (e) Robot trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s

human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane (j) Visibility graph of human 1 (k) Visibility graph of human 2 (l) Visibility graph of human 3

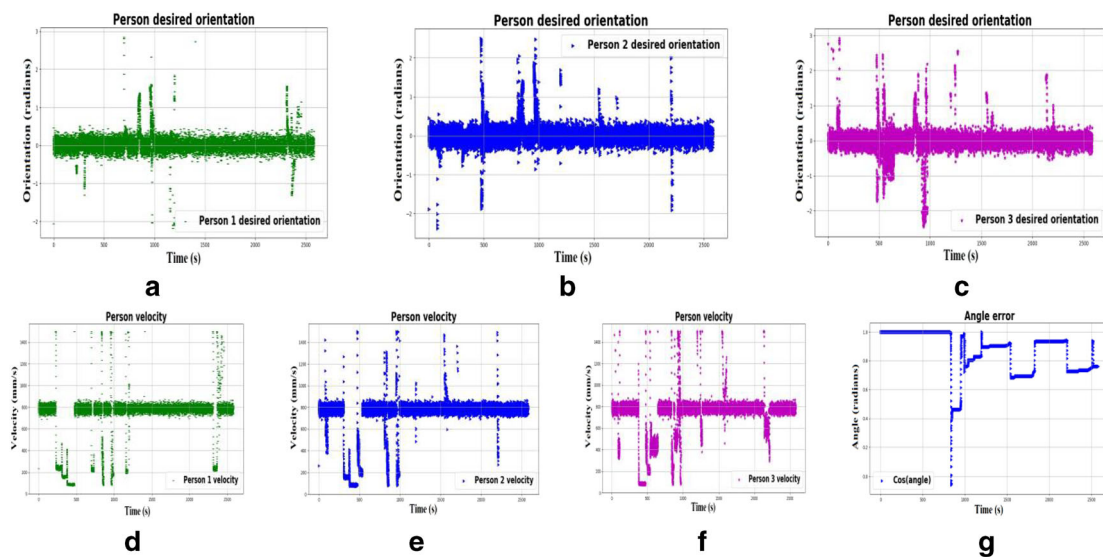


Fig. 16 Scenario ID: II Humans constraints (a) First human desired orientation (b) Second human desired orientation (c) Third human desired orientation (d) Velocity of first person (e) Velocity of second person (f) Velocity of third person (g) Corner angle error

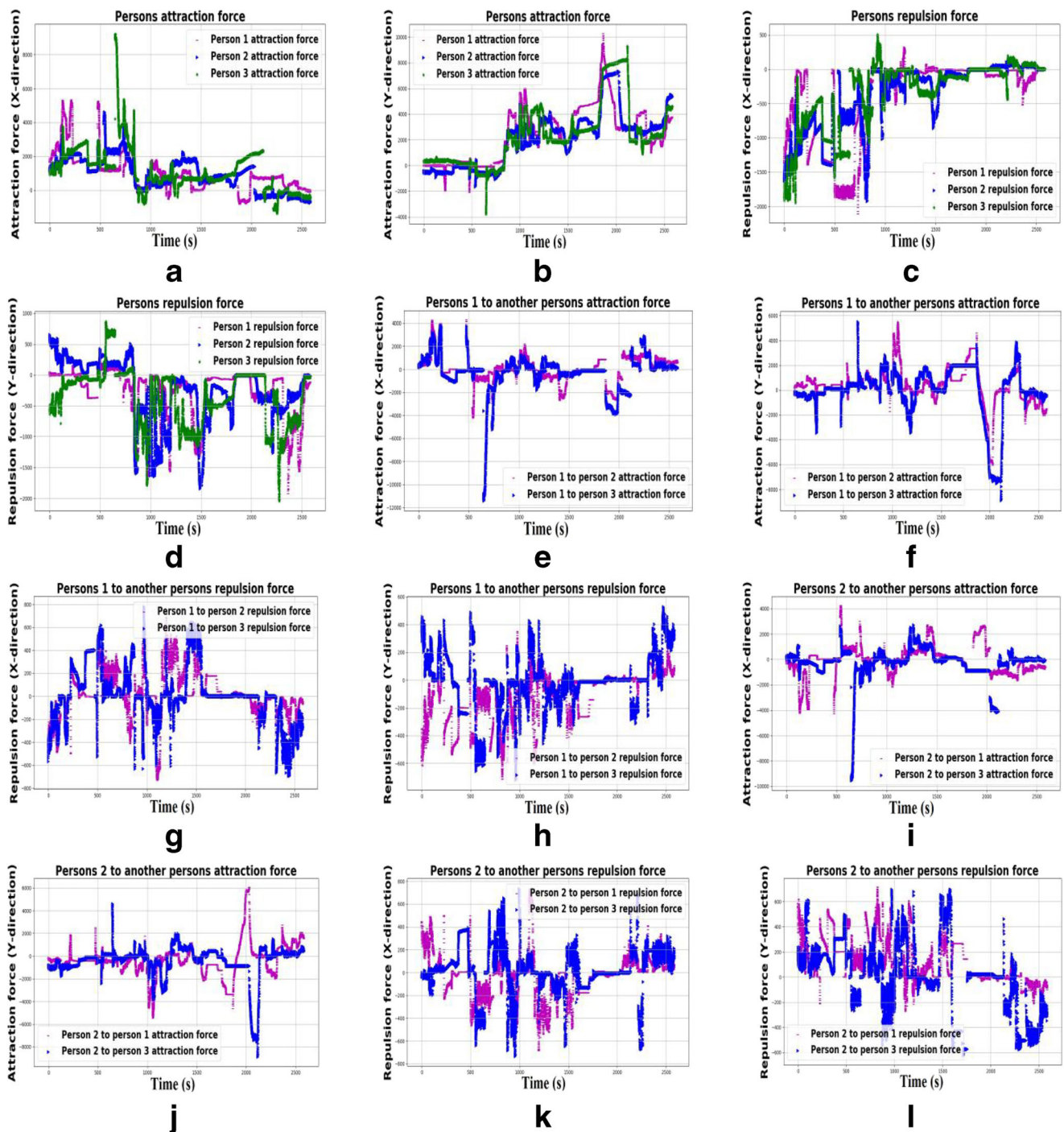


Fig. 17 Scenario ID: II Social forces modelling (a) Humans’ attraction force towards the robot in the X-direction (b) Humans’ attraction force towards the robot in the Y-direction (c) Humans’ repulsive force towards the robot in the X-direction (d) Humans’ repulsive force towards the robot in the Y-direction (e) Person 1 to another person’s attraction force in the X-direction (f) Person 1 to another person’s attraction force in the Y-direction (g) Person 1 to another person’s repulsive force in the X-direction (h) Person 1 to another person’s repulsive force in the Y-direction (i) Person 2 to another person’s attraction force in the X-direction (j) Person 2 to another person’s attraction force in the Y-direction (k) Person 2 to another person’s repulsive force in the X-direction (l) Person 2 to another person’s repulsive force in the Y-direction (m) Person 3 to another person’s attraction force in the X-direction (n) Person 3 to another person’s attraction force in the Y-direction (o) Person 3 to another person’s repulsive force in the X-direction (p) Person 3 to another person’s repulsive force in the Y-direction (q) Random force on X-plane (r) Random force on Y-plane (s) Total social force applied on human 1 in the X-direction (t) Total social force applied on human 1 in the Y-direction (u) Total social force applied on human 2 in the X-direction (v) Total social force applied on human 2 in the Y-direction (w) Total social force applied on human 3 in the X-direction (x) Total social force applied on human 3 in the Y-direction

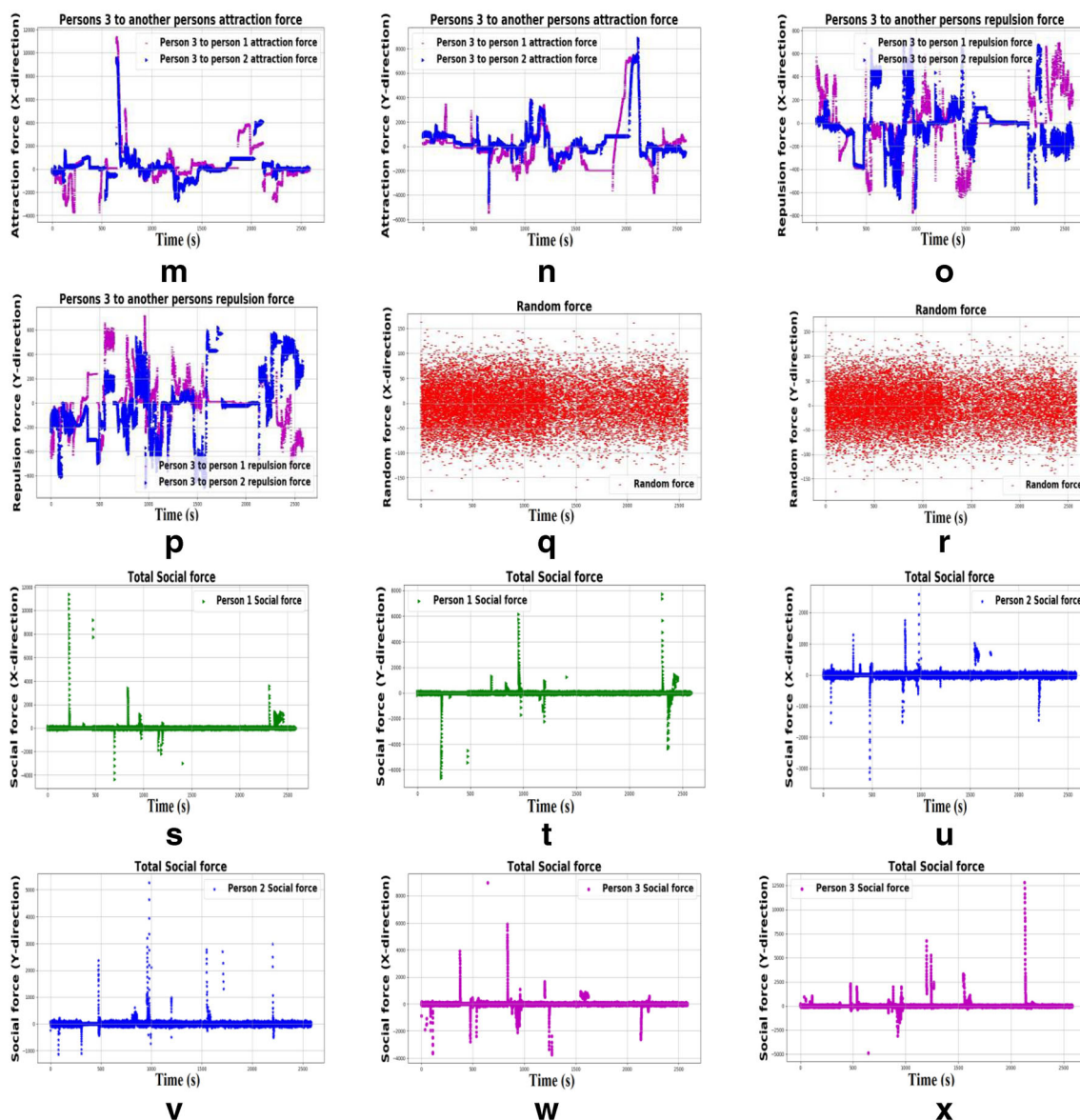


Fig. 17 (continued)

The LSTM performed poorly because it also learnt the noisy behaviours of intents to leave the group, getting distracted, etc. In our assumption, humans often act erratically, and the same behaviours cannot be filtered out. Training LSTM on the entire subject-specific makes it impossible for LSTM to decide which are the erratic behaviours and which are the genuine behaviours, and the performance is thus reasonably poor. After re-initialization, the LSTM performance improved reasonably well. However, overall, the performance was still not good due to the same reasons. The initial position given to the LSTM was itself noisy since it came from a noisy camera. Observing multiple such noisy sequences gave a

reasonably large initial error to the LSTM. The LSTM was not trained on such data and thus continued to perform even more poorly. It must be stressed that we do not intend to show that the LSTM is ineffective for solving the problem. Our intention behind the comparison is to show that the LSTM fails to perform under the settings of noisy sensor readings and highly noisy data where people often do not follow the robot. Under ideal settings, the LSTM could have given a reasonable performance.

We proposed a mechanism to detect the robot turning and use the same to anticipate the person going out of view, in which case the track was not deleted despite a loss of visibility for a prolonged time. Overall, the detection of a person leaving

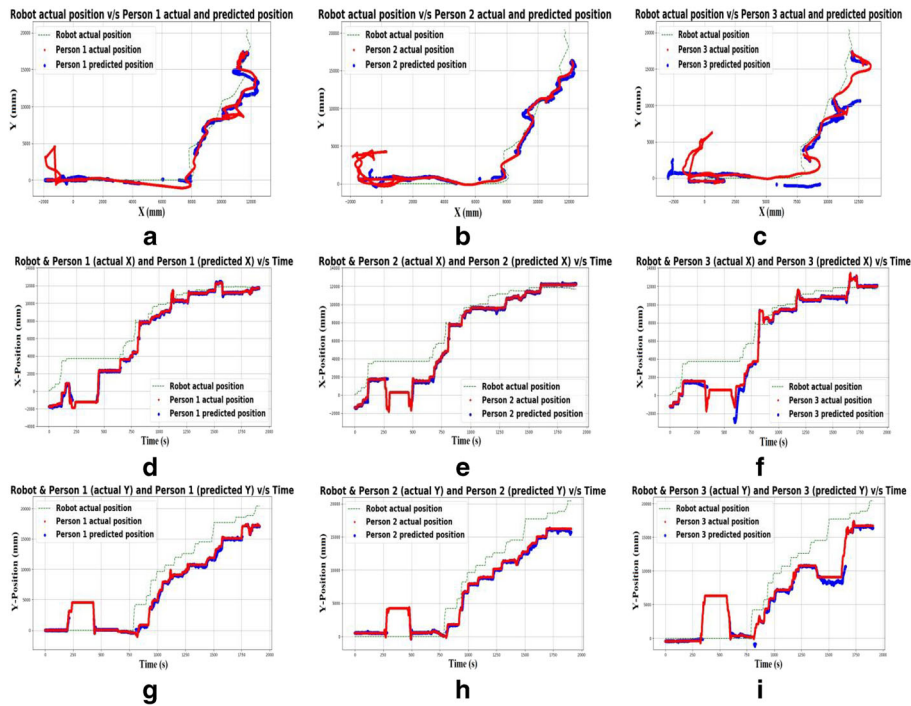


Fig. 18 Scenario ID: II 3D trajectory anticipation on a different plane using a Random particle filter (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (e) Robot trajectory v/s human 2 actual v/s anticipated trajectory on

X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

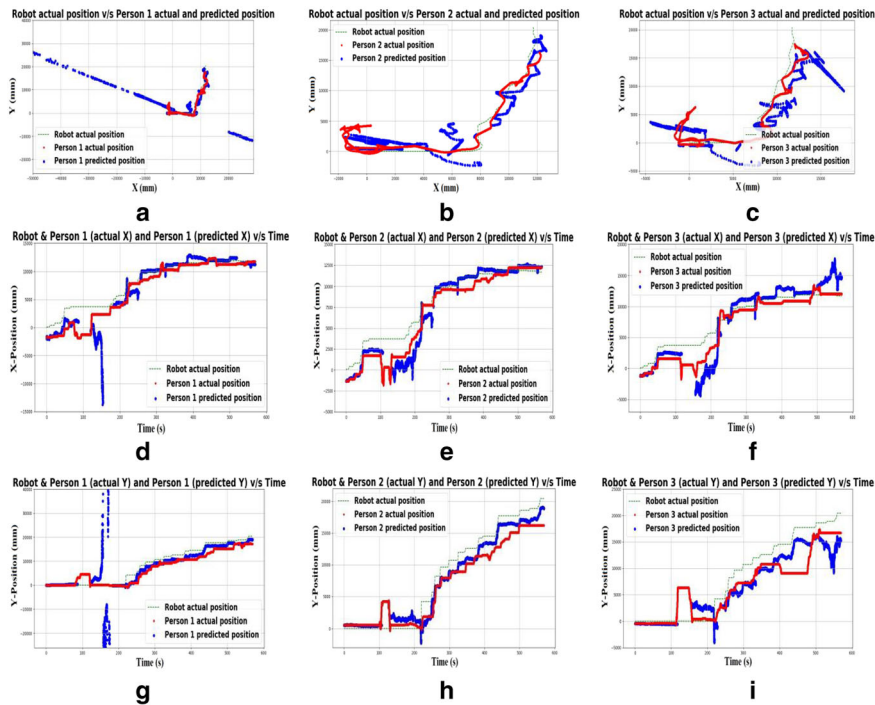


Fig. 19 Scenario ID: II 3D trajectory anticipation on a different plane using a Tracking in image (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (e) Robot

trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

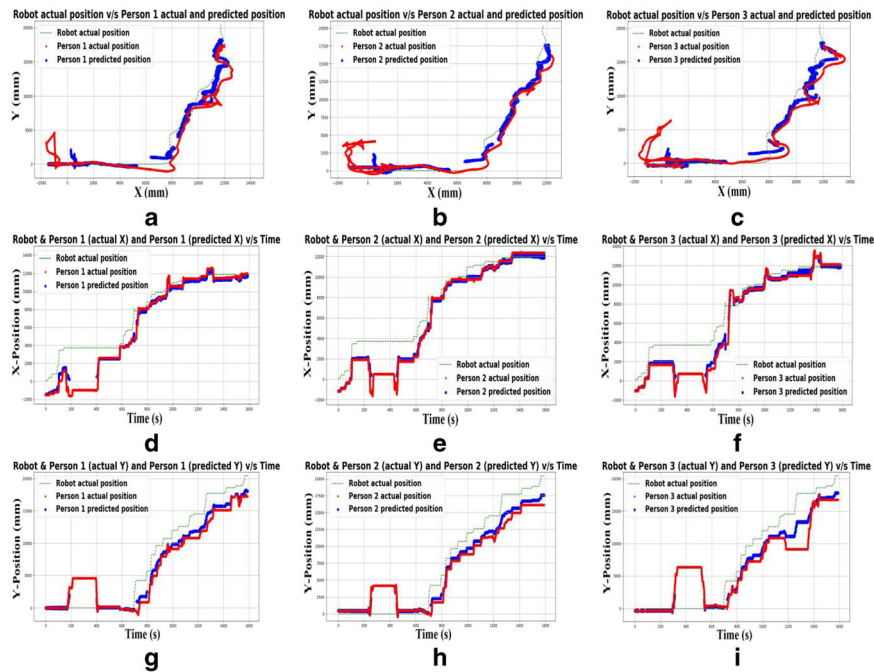


Fig. 20 Scenario ID: II 3D trajectory anticipation on a different plane using a Kalman Filter (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (e) Robot

trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

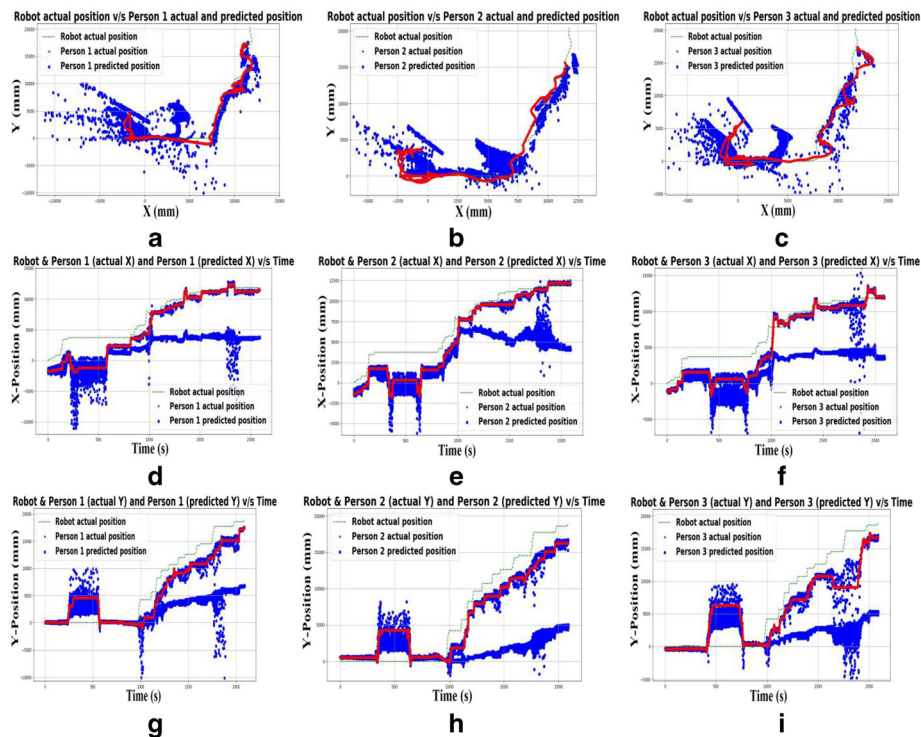


Fig. 21 Scenario ID: II 3D trajectory anticipation on a different plane using a LSTM (with re-initialization) (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane

(e) Robot trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

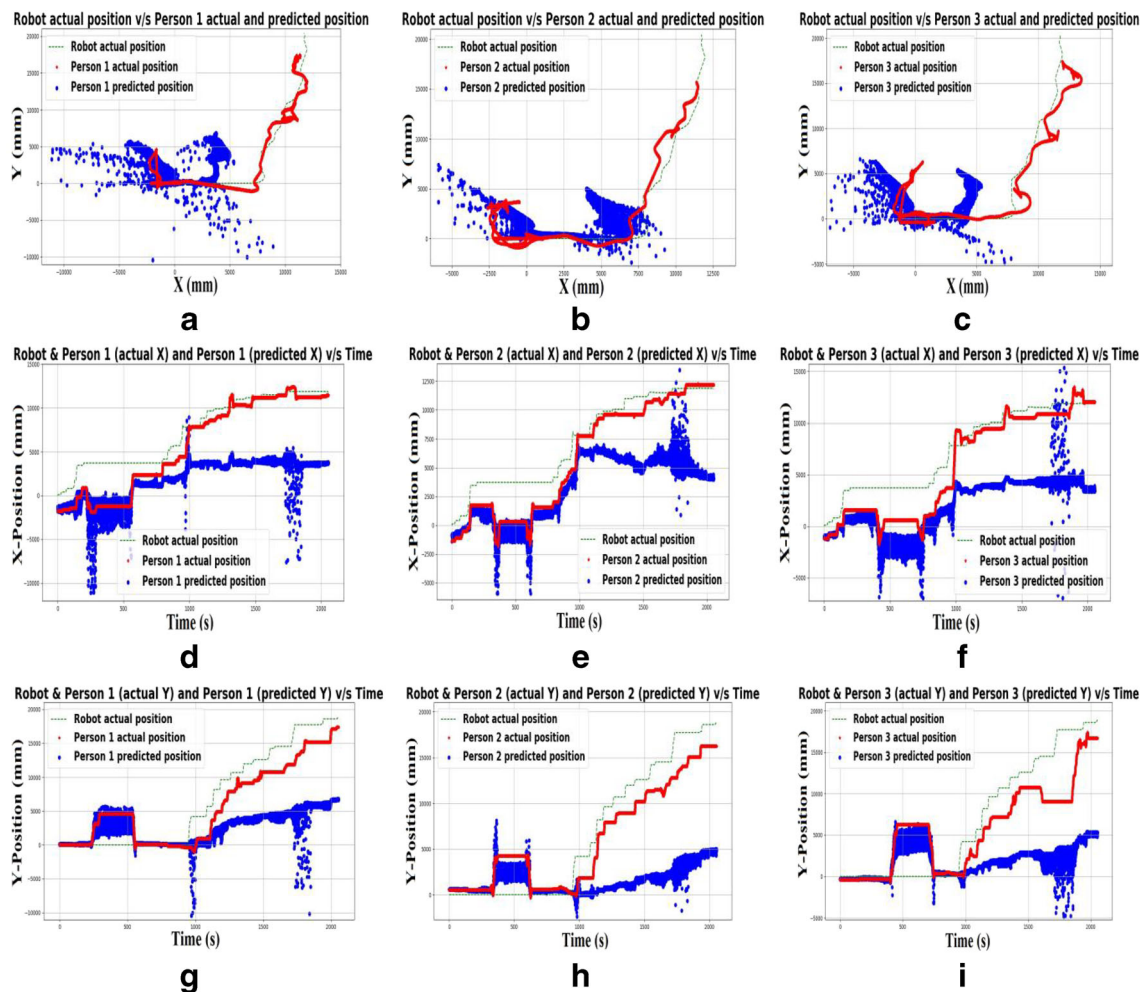


Fig. 22 Scenario ID: II 3D trajectory anticipation on a different plane using a LSTM (without re-initialization) (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (e)

Robot trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

the group was an integral part of the algorithm and hence its performance is also benchmarked. Comparisons are done with the naïve implementation of the particle filter that does not anticipate the loss of visibility due to corners and turns. F-score is used for detecting people leaving the group and the results are presented in Table 3.

The results suggest that the proposed method gives significantly better results with the corner detection module. Here the count of true positives (TP) and true negatives (TN) is increased and due to these false negatives (FN) have been reduced as shown in Table 4. Specifically, false positives (FP) have rapidly decreased and close to zero as compared to without corner detection. The proposed algorithm anticipates that since the robot is making a corner, the visibility will be lost, while without this

module the person is deleted from being tracked on almost every corner. It may be counterintuitive that this does not result in an increase in false negatives, which is because the heuristic is strong and misclassification due to the heuristic is a rarity. The little increase in true positives and true negatives; and the reduction in false positives is attributed to the accuracy of tracking. Furthermore, in Table 5 the precision, recall, and accuracy have also improved. Accuracy is 98% and above in each scenario. The precision is also high and these interpretations also prove the validity of our multi-behaviour 3D particle social force model. Verification of our model is done by the actual trajectory of the multi-agent system where we know the ground truth and the actual leaving of humans is manually annotated.

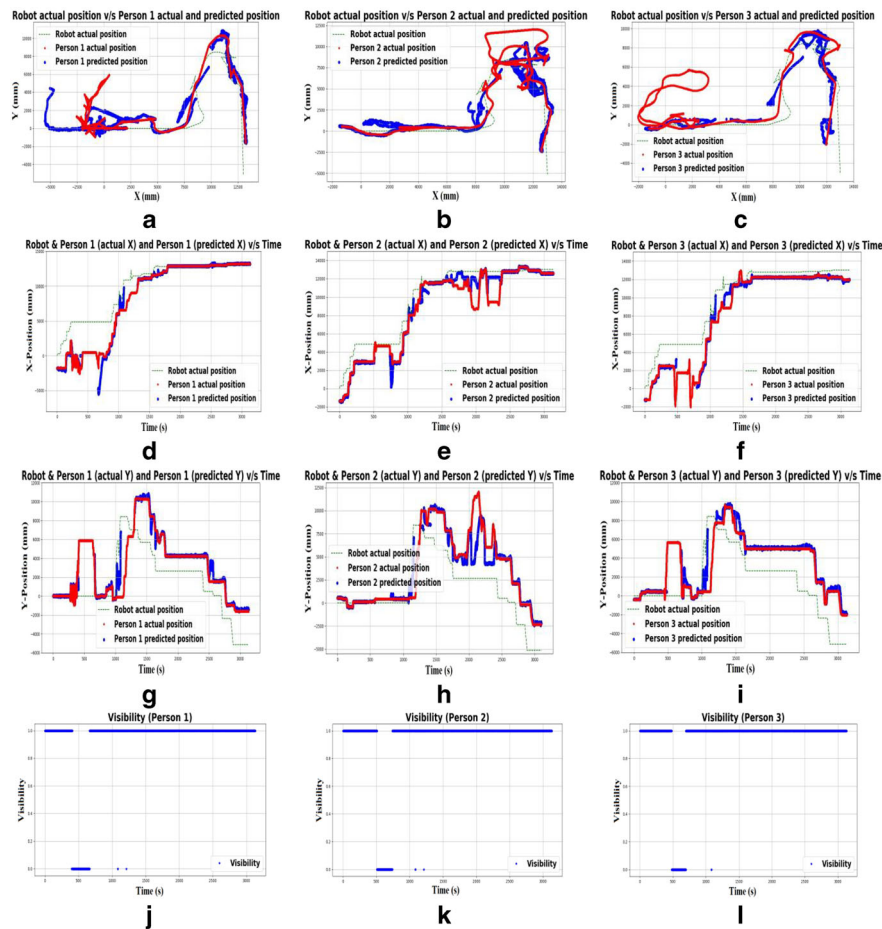


Fig. 23 Scenario ID: III 3D trajectory anticipation on a different plane using a Multi-behavioural 3D Particle Social Force Model (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (e) Robot trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s

human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane (j) Visibility graph of human 1 (k) Visibility graph of human 2 (l) Visibility graph of human 3

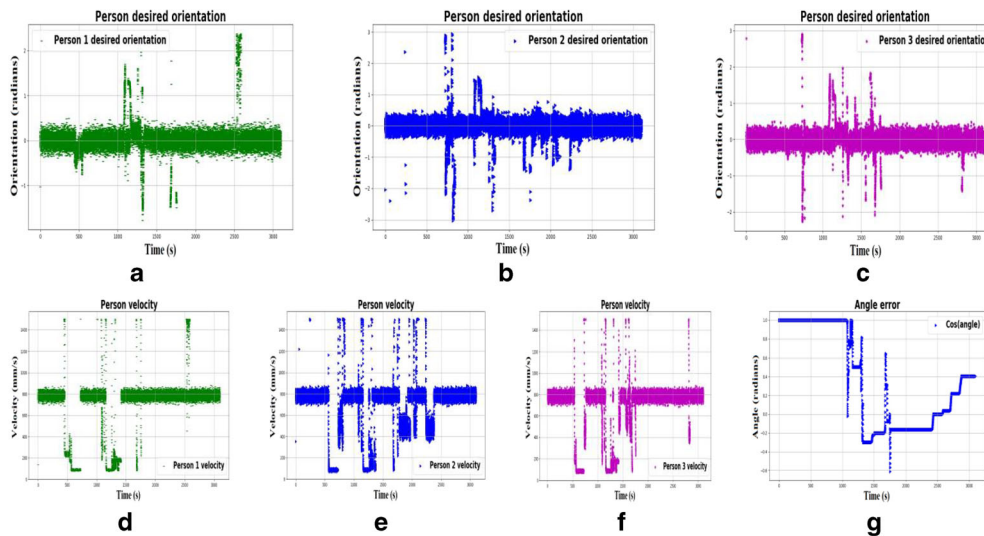


Fig. 24 Scenario ID: III Humans constraints (a) First human desired orientation (b) Second human desired orientation (c) Third human desired orientation (d) Velocity of first person (e) Velocity of second person (f) Velocity of third person (g) Corner angle error

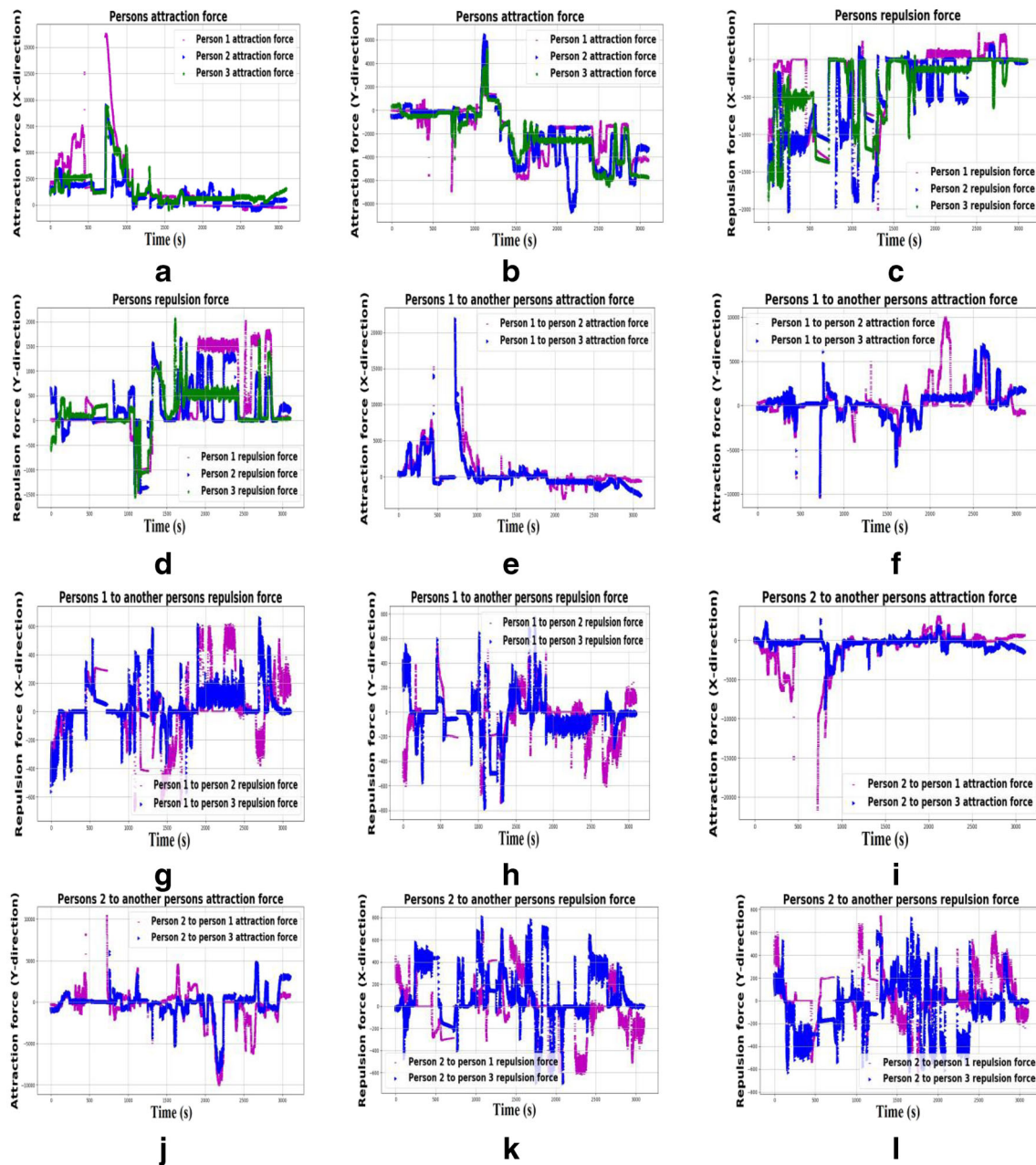


Fig. 25 Scenario ID: III Social forces modelling (a) Humans' attraction force towards the robot in the X-direction (b) Humans' attraction force towards the robot in the Y-direction (c) Humans' repulsive force towards the robot in the X-direction (d) Humans' repulsive force towards the robot in the Y-direction (e) Person 1 to another person's attraction force in the X-direction (f) Person 1 to another person's attraction force in the Y-direction (g) Person 1 to another person's repulsive force in the X-direction (h) Person 1 to another person's repulsive force in the Y-direction (i) Person 2 to another person's attraction force in the X-direction (j) Person 2 to another person's attraction force in the Y-direction (k) Person 2 to another person's repulsive force in the X-direction (l) Person 2 to

another person's repulsive force in the Y-direction (m) Person 3 to another person's attraction force in the X-direction (n) Person 3 to another person's attraction force in the Y-direction (o) Person 3 to another person's repulsive force in the X-direction (p) Person 3 to another person's repulsive force in the Y-direction (q) Random force on X-plane (r) Random force on Y-plane (s) Total social force applied on human 1 in the X-direction (t) Total social force applied on human 1 in the Y-direction (u) Total social force applied on human 2 in the X-direction (v) Total social force applied on human 2 in the Y-direction (w) Total social force applied on human 3 in the X-direction (x) Total social force applied on human 3 in the Y-direction

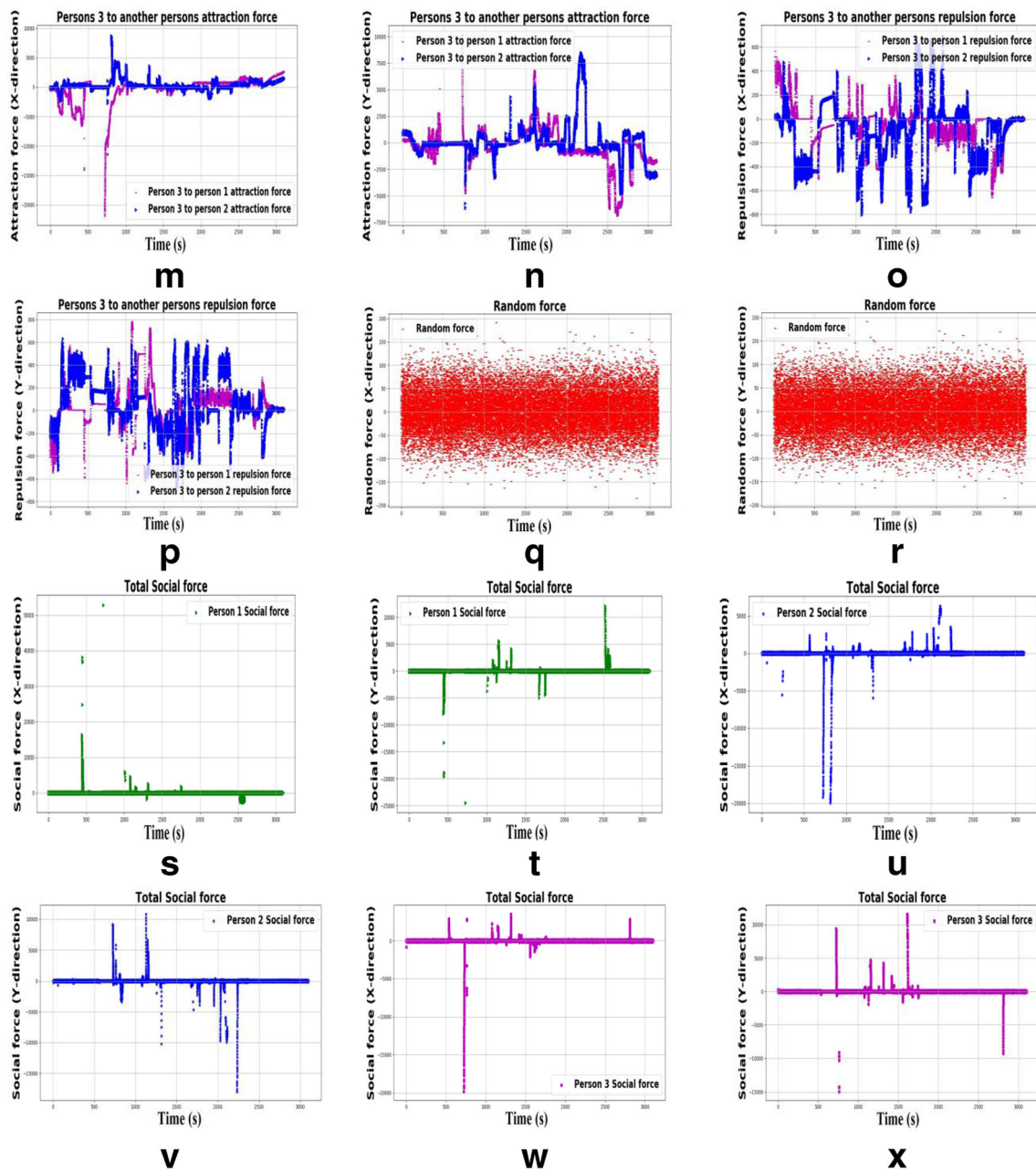


Fig. 25 (continued)

Figure 3 shows the error between the actual and the anticipated trajectory for all humans in different scenarios. Here our model renders very less error as compared to other methods. F-score is represented in Fig. 4. Multiple confusion parameters can be visualized in Fig. 5 where our model gives a good confidence score. Precision, recall, and accuracy have been represented in Fig. 6 for different scenarios and it is compared with corner and non-corner detection heuristic.

6.2 Visual representation of the anticipated trajectory and its comparison with ground truth

We applied our 3D particle model in different scenarios and each scenario elaborates the multi-behavioural humans tracking. Here the humans sometimes leave and re-join the group, swap their positions, and also overtake each other which can be visualized in the given figures. The tracker is deleted when the human leaves the group and is automatically re-initialized

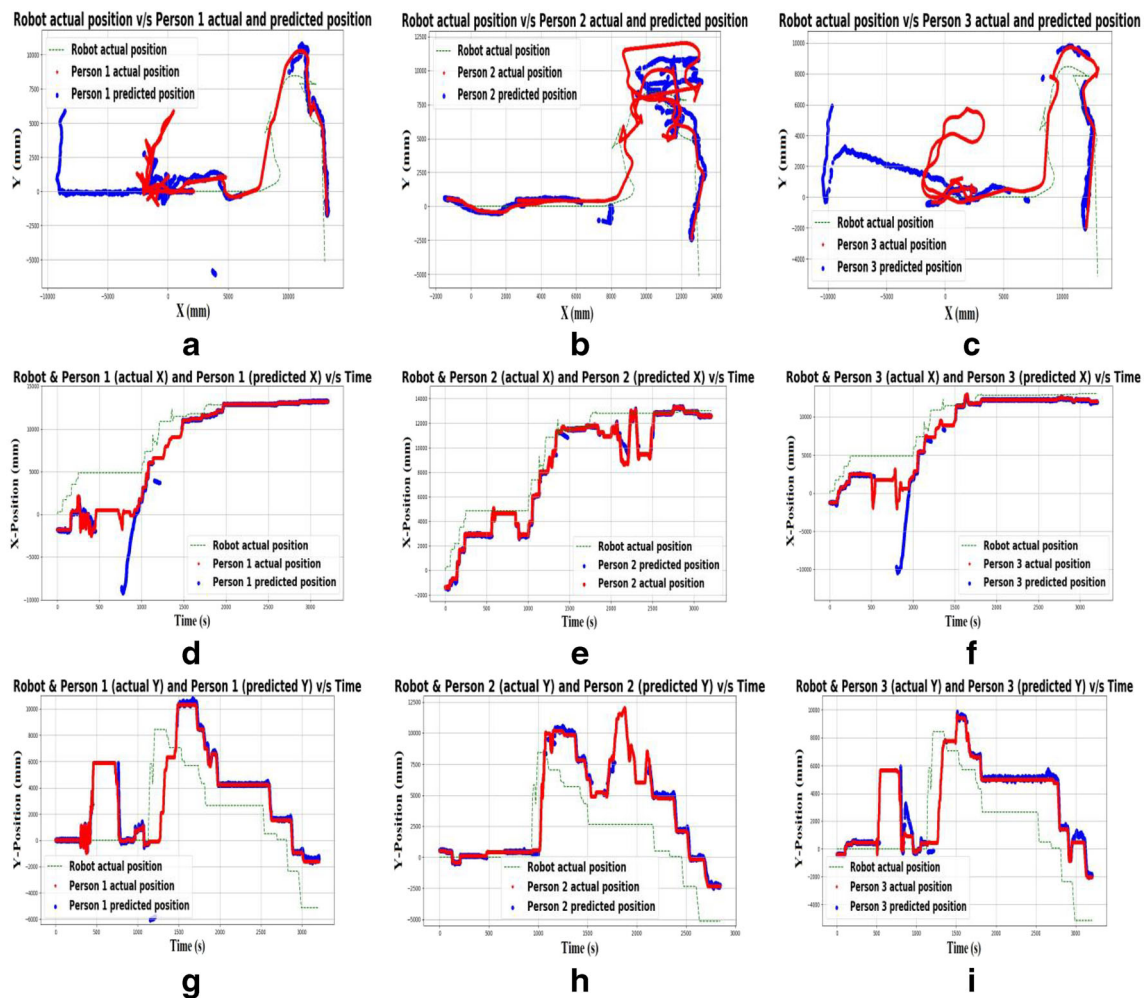


Fig. 26 Scenario ID: III 3D trajectory anticipation on a different plane using a Random particle filter (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (d) Robot trajectory v/s human 2 actual v/s anticipated

trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

when the humans re-joins the group to follow the robot. Figures 7, 15, and 23 are the graphs for the anticipated trajectory of the humans based on three different scenarios respectively using the Multi-behavioural 3D Particle Social Force Model.

The errors may seem larger than conventionally quoted in the literature for a tracker with a mobile observer model. However, three differences need to be noted in our proposed approach. First that the observer (robot) constant loses the visibility of the human. In the conventional approaches, the errors exponentially rise with time under such settings. The proposed approach does a fair deal by limiting the errors with time using the socialistic model. Second, that if the person re-joins the group, only a monocular camera is available, and the position needs to be guessed. The visibility of the person upon re-joining shall be an extremity in the image, and the cameras are very

sensitive in that region. Therefore, the re-initialization has a very large error that the tracker must overcome. Finally, the humans' teleoperators were asked to act erratically in the control of the robot. Leaving the group for some time, neither intentional nor detected by the algorithm leaves with a long sequence of no visibility where the social conventions do not hold and thus reflected upon in the results. These limitations exist in all algorithms, however, only the proposed algorithm can address these limitations sufficiently well.

The human speeds and orientation are also considered to best fit our proposed model which are represented in Figs. 8, 16, and 24 for the different scenarios. Multiple attraction and repulsive force are utilized to model this 3D social force tracking module and are shown in the given Figs. 9, 17, and 25. Tracking results based on random particle filter are drawn and explained in Fig. 10, 18,

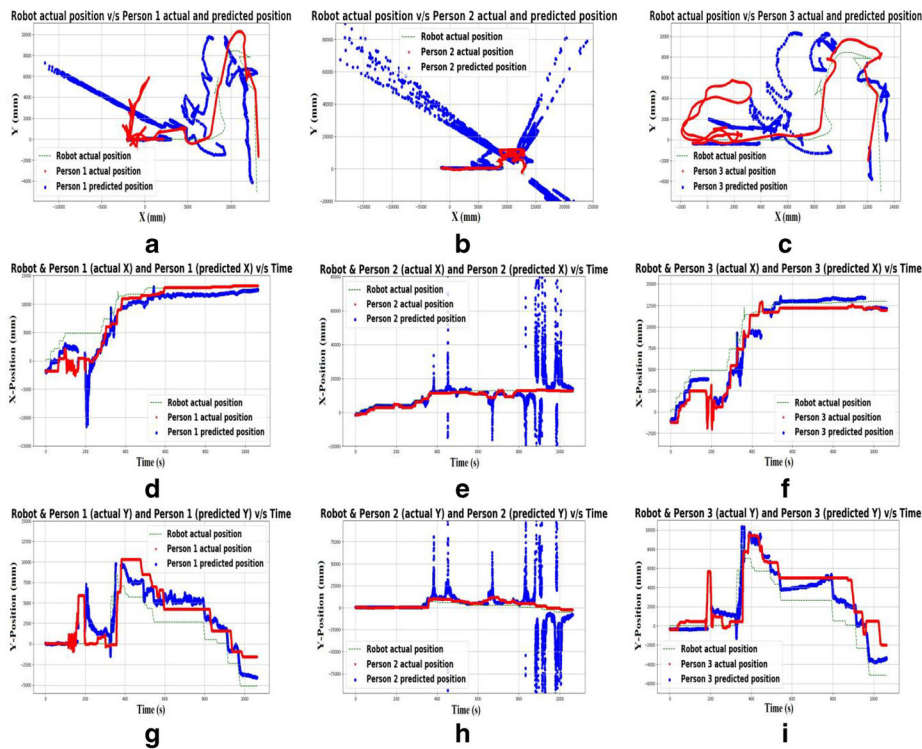


Fig. 27 Scenario ID: III 3D trajectory anticipation on a different plane using a Tracking in image (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (d) Robot

trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

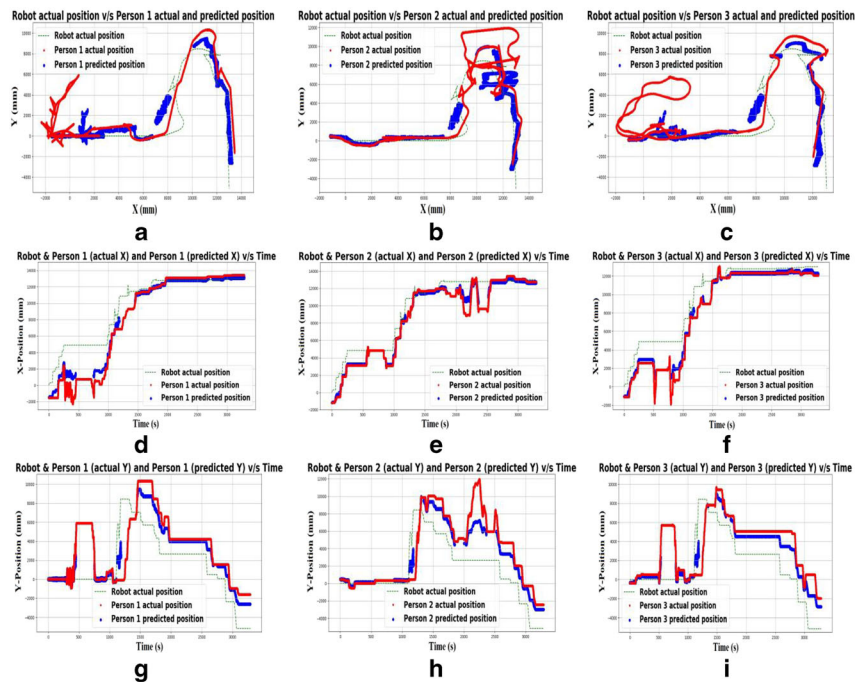


Fig. 28 Scenario ID: III 3D trajectory anticipation on a different plane using a Kalman Filter (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (d) Robot

trajectory v/s human 2 actual v/s anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

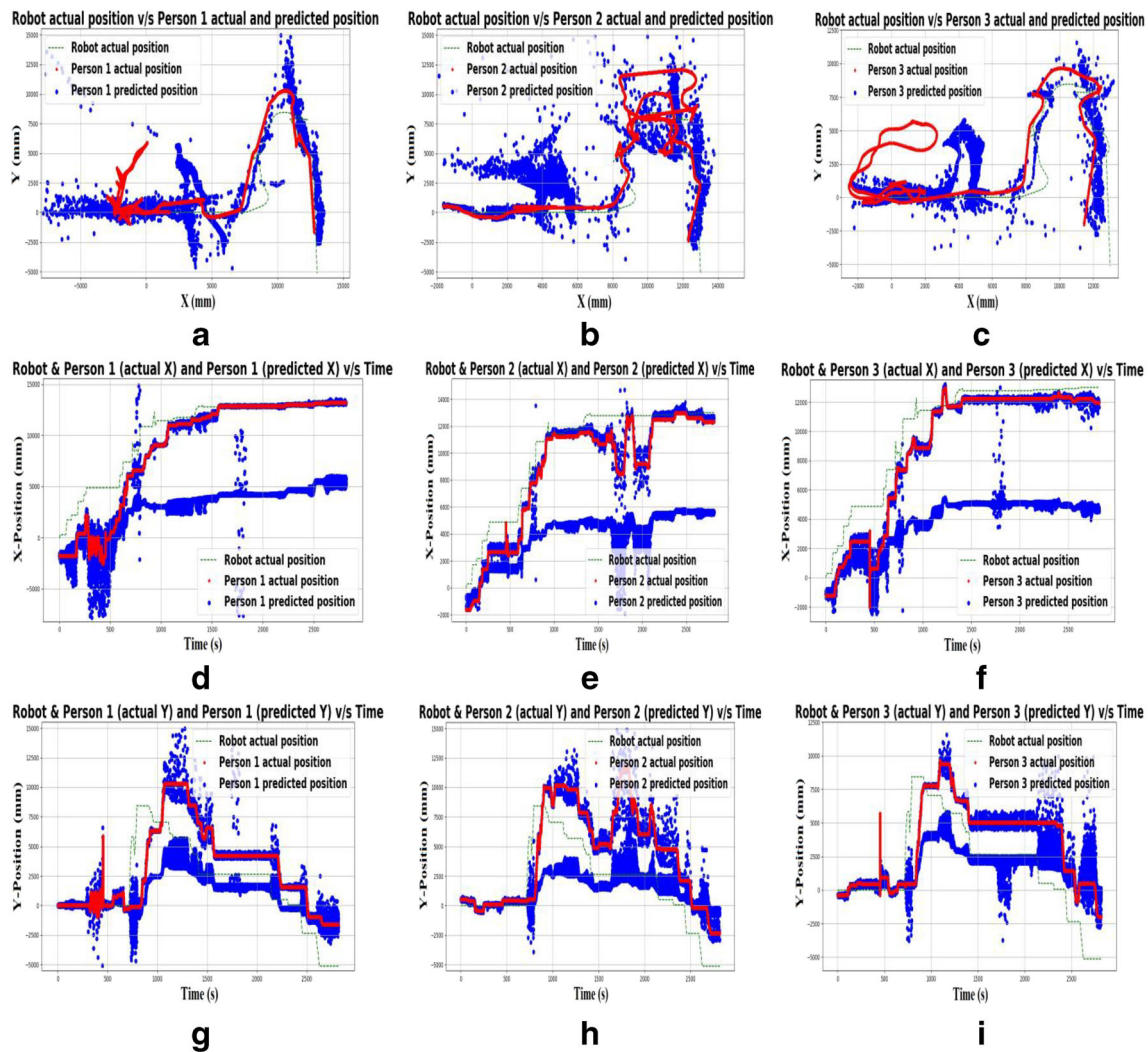


Fig. 29 Scenario ID: III 3D trajectory anticipation on a different plane using a LSTM (with re-initialization) (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (e) Robot trajectory v/s human 2 actual v/s

anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

and 26 which do not generate accurate human trajectory. Furthermore ‘tracking in image’ method is not able to track the humans in these scenarios and predicts very poor trajectories as soon as the robot is beyond visibility, which constitutes a reasonable part of the scenarios, which can be visualized in Fig. 11, 19, and 27. Additionally, we have also performed Kalman Filter approach for the trajectory prediction as shown in Figs. 12, 20, and 28 and this method is not able to anticipate the positions of the humans in no visibility or at the corners. To figure out the performance of the deep learning approach, we applied an LSTM method with re-initialization and without re-initialization and the

outcomes of LSTM with re-initialization is visualized in Figs. 13, 21, and 29 whereas the outcome of LSTM without re-initialization is visualized in Figs. 14, 22, and 30. Based on the figures it is eminent that the four methods random particle filter, tracking in the image, Kalman Filter, LSTM with re-initialization, and LSTM without re-initialization are not robust to handle the corner problem and human tracking at the corner is not possible by these approaches. Our proposed model is robust to handle these situations, can easily detect when the robot is circumventing a corner leading to no visibility and performs tracking well (Figs. 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30).

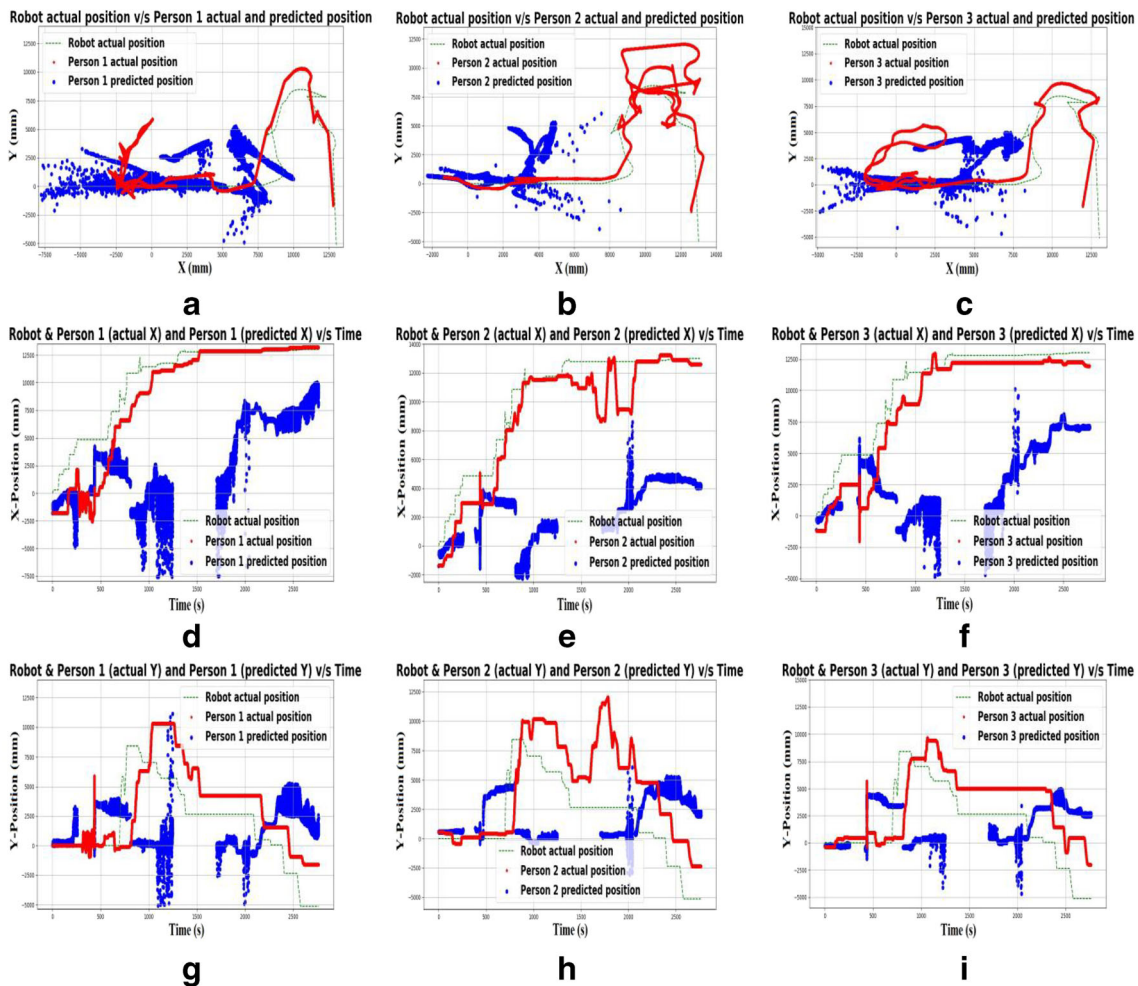


Fig. 30 Scenario ID: III 3D trajectory anticipation on a different plane using a LSTM (without re-initialization) (a) Robot trajectory v/s human 1 actual v/s anticipated trajectory (b) Robot trajectory v/s human 2 actual v/s anticipated trajectory (c) Robot trajectory v/s human 3 actual v/s anticipated trajectory (d) Robot trajectory v/s human 1 actual v/s anticipated trajectory on X-plane (e) Robot trajectory v/s human 2 actual v/s

anticipated trajectory on X-plane (f) Robot trajectory v/s human 3 actual v/s anticipated trajectory on X-plane (g) Robot trajectory v/s human 1 actual v/s anticipated trajectory on Y-plane (h) Robot trajectory v/s human 2 actual v/s anticipated trajectory on Y-plane (i) Robot trajectory v/s human 3 actual v/s anticipated trajectory on Y-plane

6.3 Real-time impression of 3D-PSF model

We have employed our 3D particle social force model for socialistic 3D tracking on multiple behaviours of humans. In Figs. 31 and 32, the different types of humans' behaviour are shown. Our model can perform tracking over different circumstances of these behaviours and at corner where the robot is taking a sharp turn and the humans are following the robot. Here the tracker is not present when any human leaves the group. As the human joins the group, the tracker is re-initialized and tracking is performed further. In Figs. 31 and 32, this phenomenon is pictured and demonstrated. Moreover, the journey started in the indoor environment and after some time the robot reaches a semi-indoor scenario where the

presence of sunlight becomes a reason for illumination change and our model performs well for tracking, although detection is not error-free here.

One of the uniqueness of the work was re-initialization once a person re-joins a group. In Fig. 31(f), one person left the group, so the track was deleted. As he joins the group again as shown in Fig. 31(g), the track is re-initialized, and tracking starts continuously. Another person again left the group and re-joined as shown in Fig. 31(i-j). In Fig. 31(k) detection did not happen for one person but tracking was continuously performed. Two persons swapped their position in Fig. 31(d-e). Further in Fig. 31(m-n), track for one person was again deleted and re-initialized. Apart from this, all the humans left the group, so the track was deleted as shown in Fig. 32(a), and as they

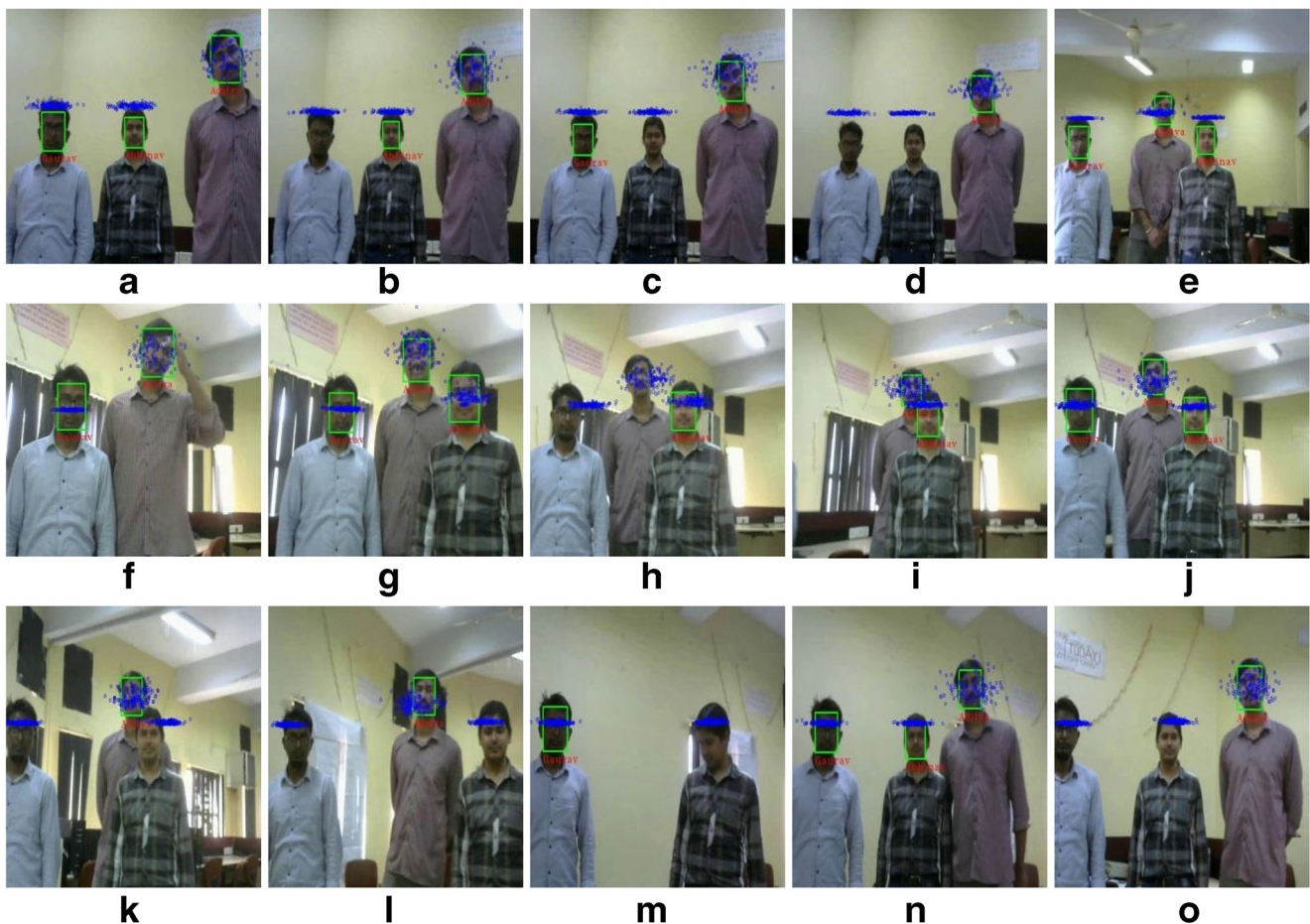


Fig. 31 Behavioural 3D tracking (a) Every person is detected and tracked by the robot. (b-d) Detection is not possible for every person, but tracking happens. (e) Two humans swapped their position and are following the robot. (f) One person leaves the group, and the tracker is deleted. (g) Person re-joins its group, and the tracker is re-uninitialized, and tracking starts. (h) A robot is taking a rotation to achieve its goal and here detection

is not possible, but the tracking model performs its job. (i-j) One person leaves the group and re-joins. (k-l) A robot continuously takes a sharp turn and humans are also tracked by the 3D tracker model. (m-n) The third person leaves the group and re-joins the group again. (o) Robot is at a corner and determines the human position in 3D

joined their group (in Fig. 32(b)), the track was re-initialized, and tracking was performed. Some more track deletion and re-initialization can be visualized in Fig. 32.

7 Conclusion

In this paper, we presented a new social-force-based particle filter that allows 3D human tracking in a group from a monocular camera placed on a mobile robot. The tracker assumes that humans follow social conventions when navigating as a group behind the robot. These behaviours are used for socialistic tracking and trajectory anticipation. The system can track humans under low visibility conditions when the robot often loses sight of the persons due to a small field of view. The system can predict a loss of

visibility due to context like traveling through a corner, which assures the robot to continue tracking while trackers are bound to delete tracks of humans in such a case. The paper thus addresses the practical problems with affordable robots operating in home and office environments alongside humans and solves the vision problem which is often the backbone of robotics.

The model was compared with baseline trackers including a particle filter, directly tracking humans in the image, LSTM without re-initialization and LSTM with re-initialization based on the image. It must be noted that tracking under limited visibility and contextualization of turns during tracking is a new problem that has not been widely addressed in the literature. Thus, no direct suitable method was available for the comparisons. The proposed model produces very less error as compared with the baseline methods on the multi-agent

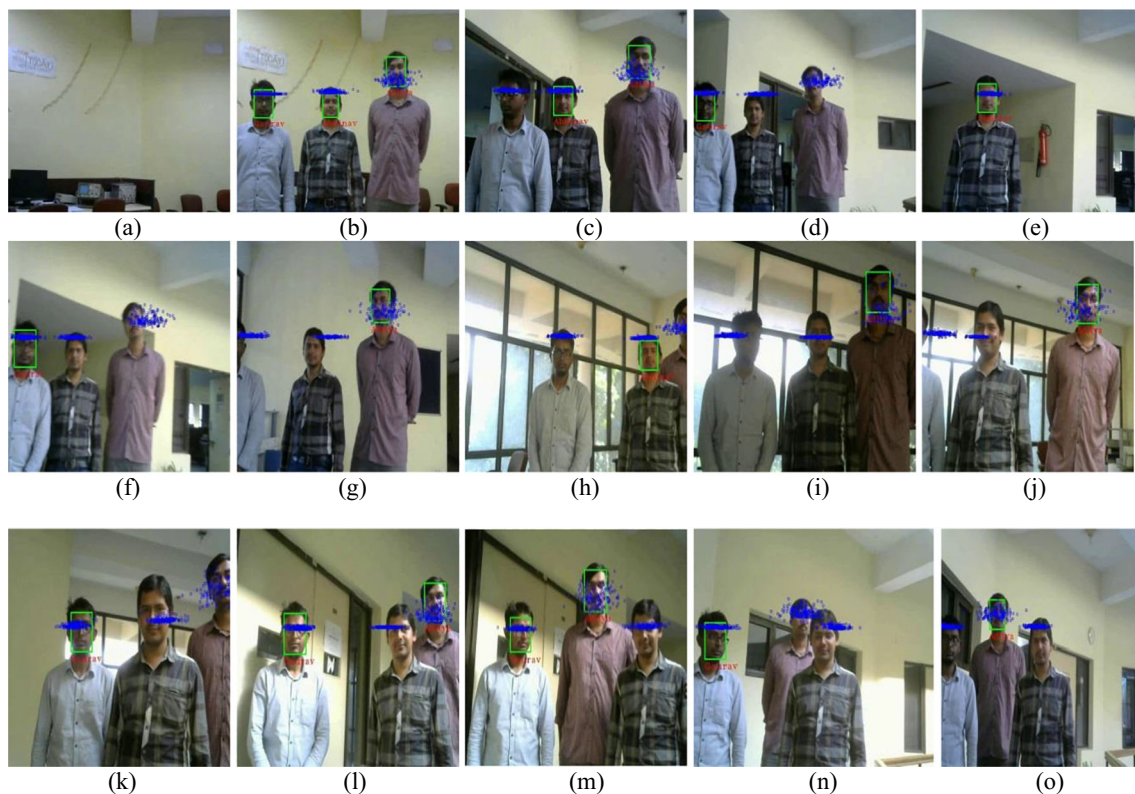


Fig. 32 Multi-behavioural 3D social tracking **(a)** All humans of the group have left and the trackers are deleted. **(b)** Everyone joins the group and trackers are available to track them. **(c)** Humans' and the robot's position at a corner. **(d)** Each human is following a robot while maintaining a queue. **(e)** Two humans have left the group, so the trackers are deleted. **(f)** Both humans join the group at the corner and follow the robot. Moreover, tracking is being performed. **(g)** Since the robot is taking a sharp turn and the face of one human is almost cropped due to the limited field of view of the camera, hence it is not detected but our model can

track in these critical circumstances. **(h)** Background illumination is changed and still tracking is performing while one human is cropped. **(i-j)** The robot is taking a sharp turn to complete its path while tracking. **(k)** One human is forward and the remaining two are backward to the one human. **(l)** One human is still backward to other ones. **(m)** Humans adjust themselves due to limited space in the corridor. **(n-o)** One human is backward to the other due to limited space on the balcony and the robot is taking a sharp turn at the corner

ground truth. An application requirement is to detect people leaving the group when their tracking must stop. False positives are rapidly decreased with the employment of our model, with no compromise on the true positives (which have a minor increase). A set of experiments with a real social robot is also done that emphasizes the appropriateness of our model to socially compliant robot navigation in real-time with a low-cost budget.

One of the major limitations of the current works is that while the experiments are done under realistic settings with the complete robotic setup including humans, the benchmarking is done using simulators to get the ground truth. Humans may not have socially simulated robots. The experiments need to be done with external trackers to prepare the ground truth. The major application was enabling the robot to take an informed decision based on the tracked positions of the people. The tracker uncertainty based active planning of the robot needs to be explored. Overall, this paper has given a

sound technique for tracking humans using low-cost sensors under limited visibility settings which leads to numerous applications that can be tried in the future.

Declarations

Conflict of interest The authors have no conflict of interest to declare.

References

1. Malviya V, Reddy AK, Kala R (2020) Autonomous social robot navigation using a behavioral finite state social machine. *Robotica* 38(12):2266–2289
2. Wang M, Liu Y, Su D, Liao Y, Shi L, Xu J, Miro JV (2018) Accurate and real-time 3-D tracking for the following robots by fusing vision and ultrasonic information. *IEEE/ASME Transactions on Mechatronics* 23(3):997–1006

3. Henriques JF, Caseiro R, Martins P, Batista J (2014) High-speed tracking with kernelized correlation filters. *IEEE Trans Pattern Anal Mach Intell* 37(3):583–596
4. Trautman P, Ma J, Murray RM, Krause A (2015) Robot navigation in dense human crowds: statistical models and experimental studies of human–robot cooperation. *Int J Robot Res* 34(3):335–356
5. Kim B, Pineau J (2016) Socially adaptive path planning in human environments using inverse reinforcement learning. *Int J Soc Robot* 8(1):51–66
6. Kretzschmar H, Spies M, Sprunk C, Burgard W (2016) Socially compliant mobile robot navigation via inverse reinforcement learning. *Int J Robot Res* 35(11):1289–1307
7. Bera A, Randhavane T, Prinja R, Manocha D (2017, September) Sociosense: robot navigation amongst pedestrians with social and psychological constraints. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 7018–7025). IEEE
8. Sung Y, Chung W (2015) Hierarchical sample-based joint probabilistic data association filter for following human legs using a mobile robot in a cluttered environment. *IEEE Trans Huma Mach Syst* 46(3):340–349
9. Dahroug B, Tamadazte B, Andreff N (2017, May) Visual servoing controller for time-invariant 3d path following with remote Centre of motion constraint. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3612–3618). IEEE
10. Chi W, Wang J, Meng MQH (2017) A gait recognition method for human following in service robots. *IEEE Trans Syst Man Cybern Syst* 48(9):1429–1440
11. Yuan J, Cai J, Zhang X, Sun Q, Sun F, Zhu W (2019) Fusing skeleton recognition with face-TLD for human following of Mobile service robots. *IEEE Trans Syst Man Cybern Syst*:1–17
12. Roozegar M, Mahjoob MJ, Esfandyari MJ, Panahi MS (2016) XCS-based reinforcement learning algorithm for motion planning of a spherical mobile robot. *Appl Intell* 45(3):736–746
13. Pereira P, Cunha R, Cabecinhas D, Silvestre C, Oliveira P (2019) A 3-D trailer approach to leader-following formation control. *IEEE Trans Control Syst Technol*
14. Kumar U, Banerjee A, Kala R (2020) Collision avoiding decentralized sorting of robotic swarm. *Appl Intell* 50(4):1316–1326
15. Paliwal SS, Kala R (2018) Maximum clearance rapid motion planning algorithm. *Robotica* 36(6):882–903
16. Wang A, Steinfeld A (2020) Group Split and merge prediction with 3D convolutional networks. *IEEE Robot Automation Lett* 5(2):1923–1930
17. Jeddisaravi K, Alitappeh RJ, Pimenta LC, Guimaraes FG (2016) Multi-objective approach for robot motion planning in search tasks. *Appl Intell* 45(2):305–321
18. Yu Y, Mora KAF, Odobez JM (2018) HeadFusion: 360° head pose tracking combining 3D Morphable model and 3D reconstruction. *IEEE Trans Pattern Anal Mach Intell* 40(11):2653–2667
19. Malviya V, Kala R (2018, October) Tracking vehicle and faces: towards socialistic assessment of human behaviour. In *2018 Conference on Information and Communication Technology (CICT)* (pp. 1–6). IEEE
20. Tang H, Sun W, Yu H, Lin A, Xue M, Song Y (2019) A novel hybrid algorithm based on PSO and FOA for target searching in unknown environments. *Appl Intell* 49(7):2603–2622
21. Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process Lett* 23(10):1499–1503
22. Li H, Lin Z, Shen X, Brandt J, Hua G (2015) A convolutional neural network cascade for face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5325–5334)
23. Khaksar W, Hong TS, Khaksar M, Motlagh O (2014) A fuzzy-tabu real time controller for sampling-based motion planning in unknown environment. *Appl Intell* 41(3):870–886
24. Thabet E, Khalid F, Sulaiman PS, Yaakob R (2018) Fast marching method and modified features fusion in enhanced dynamic hand gesture segmentation and detection method under complicated background. *J Ambient Intell Humaniz Comput* 9(3):755–769
25. Tian S, Yin XC, Su Y, Hao HW (2017) A unified framework for tracking based text detection and recognition from web videos. *IEEE Trans Pattern Anal Mach Intell* 40(3):542–554
26. Pozna C, Troester F, Precup RE, Tar JK, Preitl S (2009) On the design of an obstacle avoiding trajectory: method and simulation. *Math Comput Simul* 79(7):2211–2226
27. Haidegger T, Kovács L, Precup RE, Preitl S, Benyó B, Benyó Z (2011) Cascade control for telerobotic systems serving space medicine. *IFAC Proceedings* 44(1):3759–3764
28. Fiorini L, Mancipopi G, Semeraro F, Fujita H, Cavallo F (2020) Unsupervised emotional state classification through physiological parameters for social robotics applications. *Knowl-Based Syst* 190:105217
29. Sun L, Yan Z, Mellado SM, Hanheide M, Duckett T (2018, May) 3DOF pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1–7). IEEE
30. Kumar N, Sukavanam N (2020) An improved CNN framework for detecting and tracking human body in unconstraint environment. *Knowl-Based Syst* 193:105198
31. Xie G, Gao H, Qian L, Huang B, Li K, Wang J (2017) Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models. *IEEE Trans Ind Electron* 65(7):5999–6008
32. Marchetti F, Becattini F, Seidenari L, Del Bimbo A (2020) Multiple trajectory prediction of moving agents with memory augmented networks. *IEEE Trans Pattern Anal Mach Intell*:1
33. Reddy AK, Malviya V, Kala R (2020) Social cues in the autonomous navigation of indoor Mobile robots. *International Journal of Social Robotics*, 1–24
34. Bencherif A, Chouireb F (2019) A recurrent TSK interval type-2 fuzzy neural networks control with online structure and parameter learning for mobile robot trajectory tracking. *Appl Intell* 49(11):3881–3893
35. Bevilacqua P, Frego M, Fontanelli D, Palopoli L (2018) Reactive planning for assistive robots. *IEEE Robot Automation Lett* 3(2):1276–1283
36. Jang S, Elmqvist N, Ramani K (2015) Motionflow: visual abstraction and aggregation of sequential patterns in human motion tracking data. *IEEE Trans Vis Comput Graph* 22(1):21–30
37. Liu Z, Huang J, Han J, Bu S, Lv J (2016) Human motion tracking by multiple RGBD cameras. *IEEE Trans Circuits Syst Video Technol* 27(9):2014–2027
38. Castillo JC, Fernández-Caballero A, Serrano-Cuerda J, López MT, Martínez-Rodrigo A (2017) Smart environment architecture for robust people detection by infrared and visible video fusion. *J Ambient Intell Humaniz Comput* 8(2):223–237
39. Choi C, Christensen HI (2012) Robust 3D visual tracking using particle filtering on the special Euclidean group: a combined approach of keypoint and edge features. *Int J Robot Res* 31(4):498–519
40. Choi C, Christensen HI (2011, May) Robust 3D visual tracking using particle filtering on the SE (3) group. In *2011 IEEE*

- International Conference on Robotics and Automation (pp. 4384-4390). IEEE
41. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823)
 42. Wen Y, Zhang K, Li Z, Qiao Y (2016, October) A discriminative feature learning approach for deep face recognition. In European conference on computer vision (pp. 499-515). Springer, Cham

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Vaibhav Malviya received the M.Tech degree in Computer Science & Engineering from Jaypee Institute of Information Technology, Noida, India in 2015 and he is currently pursuing Ph.D from the Indian Institute of Information Technology Allahabad (Prayagraj), India. His research interests are vision application for robotics and robot motion planning.



Rahul Kala received the B.Tech. and M.Tech. degrees in Information Technology from the Indian Institute of Information Technology and Management, Gwalior, India in 2010. He received his Ph.D. degree in cybernetics from the University of Reading, UK in 2013. He is currently working as an Assistant Professor in the Indian Institute of Information Technology, Allahabad, India in the Centre for Intelligent Robotics. He is the author of four books and over 100 papers. He has numerous funded projects from Government agencies and industry to his name. He is a recipient of the Early Career Research Grant from the Department of Science and Technology, Government of India. He is a recipient of the Best PhD dissertation award from the IEEE Intelligent Transportation Systems Society; and a scholarship under the Commonwealth Scholarship and Fellowship Program from the UK Government.