Robotica: page 1 of 24. © The Author(s) 2020. Published by Cambridge University Press.
doi:10.1017/S0263574720000259

Autonomous Social Robot Navigation using a Behavioral Finite State Social Machine

Vaibhav Malviya*, Arun Kumar Reddy and Rahul Kala

Centre of Intelligent Robotics, Indian Institute of Information Technology, Allahabad, Prayagraj, India. E-mails: arunkumarreddy1997@gmail.com, rkala001@gmail.com

(Accepted March 26, 2020)

SUMMARY
We present a robot navigation system based on Behavioral Finite State Social Machine. The paper makes a robot operate as a social tour guide that adapts its navigation based on the behavior of the visitors. The problem of a robot leading a human group with a limited field-of-view vision is relatively untouched in the literature. Uncertainties arise when the visitors are not visible, wherein the behavior of the robot is adapted as a social response. Artificial potential field is used for local planning, and a velocity manager sets the speed disproportional to time duration of missing visitors.

KEYWORDS: Socialistic behavior; Artificial potential field; Behavioral Finite State Machine; Autonomous navigation; Social robot motion planning; Service robotics.

List of Symbols and Acronyms

Symbols	Description
Positions	
$q_R(x_R, y_R, 0, \theta_R)$	Robot's pose in the real world (X, Y, Z, orientation)
$q_C(x_C, y_C, h_C, \theta_R + \pi)$	Pose of the rear camera at the robot (X, Y, Z, orientation)
$q_V(x_V, y_V, h_V, \theta_V)$	Visitor's face position (X, Y, Z, orientation)
q_I	Projection of visitor's face on the image
v_V, ω_V	Linear and angular speed of the visitor
I	Image
F	All faces seen in a frame
F	A single face
width(f)	Width of the face
height(f)	Height of the face
length()	Length (number of elements) of an array
Transforms	
T_R^C	Transformation between the robot and the camera
$T_V^R(q_V, q_R)$	Transformation between the visitor and the robot

* Corresponding author. E-mail: vaibsidea@gmail.com

ζ	Distortion corrected calibration matrix
$f_V\ (q_I)$	Visibility function, is q_I within the field of view of the camera?

Motion to goal

V	Ordered sequence of sites
v_i	i^{th} visit site
$v_i.q$	Location of the site v_i
$v_i.t$	The information to be announced about site v_i
$v_i.\tau_d$	Deliberative trajectory to reach the visit site v_i
$s_{i,j}$	j^{th} sub-goal of visit site i
$g\ (x_g,y_g)$	Goal or next visit site (X, Y)
$s\ (x_s,y_s)$	current sub-goal in pursuit of goal (X, Y)
θ_s	Angle from robot to sub-goal
K_1	Gain constant for linear speed
K_2	Gain constant for angular speed
d()	Distance function
r	Sum of radii of the robot and visitor and a little extra safety distance
T	Current time
v_R^{pref}	Maximum preferred speed
v^{max}	Maximum allowable operating speed of the robot
E	Accepted tolerance for the goal
E	Accepted tolerance for the sub-goal.

Behavioral Finite State Social Machine

$t_{lastSeenV}$	Time at which the visitor V was last seen
t_{stop}	Maximum time a visitor can be unseen, after which a visitor is said to have left the group
$t_{announce}$	Time after which an announcement is made if the visitor is still unseen
$t_{stopped}$	Time at which the robot was last stopped
n	Number of visitors in group
$[\eta^{min}{}_x, \eta^{min}{}_y]$	Expected face size in the image for a socialistic distance
$left_V$	Has visitor V temporarily left the group?
$P\ (left_V)$	Probability that the visitor V has temporarily left the group
$turn$	Is robot making a sharp turn
θ_e	Orientation error of robot to goal
η	Constant beyond which a robot is labeled as turning

Artificial potential field

$[\phi_{min}, \phi_{max}]$	Field of view of the laser sensor
d_φ	Distance reported by laser sensor at an angle φ
α	Repulsive constant
β	Attractive constant
F_{rep}	Repulsive force
F_{att}	Attractive force
F_{total}	Total force

Face detection

c_i	Sample data
M_i^{det}	Cross-entropy loss of detection

n_{i}^{det}	Ground truth label of detection
P_{i}	Network-obtained probability of detection
M_{i}^{box}	Euclidean loss on bounding box prediction
n_{i}^{box}	Predicted bounding box
n_{i}^{box}	Ground truth coordinate of bounding box prediction
R^{d}	Set of real number to the power d
$M_{\mathrm{i}}^{landmark}$	Euclidean loss on facial landmark prediction
$n_{\mathrm{i}}^{landmark}$	Predicted facial landmark coordinates
$n_{\mathrm{i}}^{landmark}$	Ground truth coordinates of facial landmark
N'	Number of training samples
$\aleph_{\mathrm{j}}$	Task importance
$\beth_{\mathrm{i}}^{j}$	Sample-type indicator

1. Introduction

In the recent years, the robotics field has gained more attention, and the different wings of social robotics have also created the awareness of many researchers who have presented various applications like robots companion,[1] people evacuation[2] and cooperative exploration.[3] There are challenging and interesting issues which are involved in the domain of cooperative and social robotics, including uncertainty in the manner by which the humans react to a robot, inability to predict the human behavior, inability to simulate human behavior for rapid testing. In this paper, we consider the application of navigating visitors in a group using a socialistic and cooperative robot as a tour guide. In this paper, we develop an architectural module for guiding visitors in a group with the robot. Generally, if visitors are moving in a museum or a shopping mall, they do not have any idea of the correct tour path. It may not be possible to employ human guides for monetary reasons. The robots know the correct path of the place and can guide all the visitors in a group. In this application, the robot detects the visitors by their faces. Initially, the robot detects the visitors by their faces, and if all the visitors of a group are physically present and detected by the robot, then the robot starts its tour and navigates all the visitors toward different visit sites. As the robot reaches the visit sites, it stops and explains the interesting facts of that place and then moves toward its next visiting site. If visitors are missing, the robot must reach socially, which is the problem handled in this paper.

The current works in social robot motion planning have not appreciably looked into the specific behavior where the humans follow a leader robot, while the works largely concentrate on socially avoiding the people by a robot or a robot following the people. The major challenge that separates this problem from the rest of the variants is that the people can show noncooperative and strange behaviors like missing out at times for long durations and the robot must still socially adapt the behavior without showing strong gestures. The other major challenge is that such implementations have a large number of corner cases considering the limited field of view of the robot. Unlike other variants, looking at the group at the back, while navigating forward is not an intuitive human behavior, and therefore the general heuristics of the social motion planning applications fail to recognize the difference in needs in the modeling of this specific behavior. Consider the example of a robot turning at a sharp corner. Now the person will be out of view of the camera as the robot makes the sharp turn, and if the robot assumes that the person is missing, it will end up waiting at a sharp corner and thus causing a blockage. Similarly, consider the person briefly leaves the group. Stopping or asking the person to rejoin may both be impolite.

The paper uses a Behavioral Finite State Social Machine (BFSSM) modeling to make the robot react to the uncertainties of the human and to socially take the entire group with it, stopping and announcing if absolutely necessary. The modeling technique uses discrete states to denote the robot controller currently active. The robot controller or behavior may change based on some transition conditions. Unlike a conventional finite state machine, every state represents a controller or continuous dynamics within the state, invoked by the controller. Social force and artificial potential field (APF) have been used for the physical motion of the robot. The individual behaviors modeled are inspired from the decision-making of human tour leaders, and the robot judiciously changes its behaviors as per the context. Furthermore, the robot can anticipate the person moving outside field of view and make decisions accordingly. This is the first such attempt to model this critical leading behavior by the robot under limited field-of-view settings to the best of the knowledge of the authors.

Motion of humans can be illustrated as if the humans are subjected with social forces by each other. The model of human behavior has been studied by measurements of position, velocity, flow, etc and forms a basis to design a social robot behavior. The human behavior is chaotic and not easy to predict, especially in complex situations, which makes it hard to model the behavior. It is assumed that the human generally move in the same direction; and they fluctuate or suddenly change direction toward goal at narrow crossing areas. These circumstances happen due to nonlinear interaction among the human in a group. The problem of finding the potential force to achieve the given behavior is a very challenging and open issue and mostly non-decidable.

A hierarchical methodology[4] has evolved to map and model the potential force to achieve the given behavior of the human. Potential laws are planned where intra-group social force is defined among individuals of each robot group and also the inter-group social force between individuals of different groups. Dynamic sequence potential rules are modeled which are changed over the time at discrete intervals. Usually, the force has two constituents, attraction and repulsion. Attraction dominates when the robots are distant to each other, while repulsion dominates if the robots are too close.

Here, we have developed a framework for navigation of visitors in a group as briefly shown in Fig. 1. Visitors are detected by their faces, and as all visitors are detected by the robot, the robot continues the tour. While the robot is navigating, all the visitors should follow the robot. If any visitor misses or does not follow the robot, then the robot de-accelerates and decreases its velocity from current velocity to zero. This marks that the robot is not sure whether the missing visitor cannot be seen due to uncertainty, the missing visitor cannot be seen as it is out of field of view, or the visitor is just temporarily missing by will. Slowing down is similar to the human gesture of letting the visitor catch up. Consequently, if the missing visitor is again detected by the robot while slowing down, then again the robot accelerates and continues the navigation. However, if the missing visitor is not found for a long time, the robot is sure that the visitor is lost or has left the group without intimation. In such a case, the robot will make an announcement for the missing visitors, after the visitors are missing for sufficient time. So if the visitors miss for a long duration of time, then the robot stops and waits for the visitors. If the visitors appear and are detected by the robot, then the robot sends an acknowledgment thorough an announcement and continues the tour. Robot moves to the next visit site position, stops there and explains the site to the visitors.

In the path of the robot, there will be numerous sharp turns because workspaces are designed for humans who generally have no problems in making swift turns at walking speeds. Generally, if the robot is navigating the visitors in a museum, shopping mall, virtual or historical place, etc., then the path of the way is not always as smooth; there are lots of sharp turns especially around corners and when the robot needs to avoid an oncoming person. Such turns cause a problem for the robot. When the robot is encountering a sharp turn, it will not be able to see the visitors who will be outside the field of view. A naive controller may interpret that the visitors are missing and may make the robot stop. This complicates the problem as the robot may stop at a sharp corner which is not wide enough to accommodate visitors to the rear. However, the robot must realize that the visitors are outside the field of view instead. Similarly, even after the robot has successfully completed the sharp turn, the visitors following will take time to account for the same turn and it may undesirable for the robot to stop.

Conventional social robot applications assume that perfect information of the humans is available through sensors. This is not always possible due to sensor noise, limited field of view and sudden changes in the robot's/visitor's orientation. The natural response to the uncertainties in the literature is to make better tracking algorithms, which makes the tracking and mapping modules independent to navigation decision-making. The major novelty in this approach is that the two components are knit together, wherein the robot socially responds to the situation so as to mitigate uncertainties in cooperation with the humans. This happens by first slowing down, giving a soft signal to the human; thereafter stopping and subsequently making an announcement, letting the human know the problem. The human then cooperates and comes within the view to mitigate uncertainties.

The second novelty of the paper is using the contextual information from the navigation modules to reason for uncertainties at the tracking and mapping levels. When the robot makes a sharp turn, it is

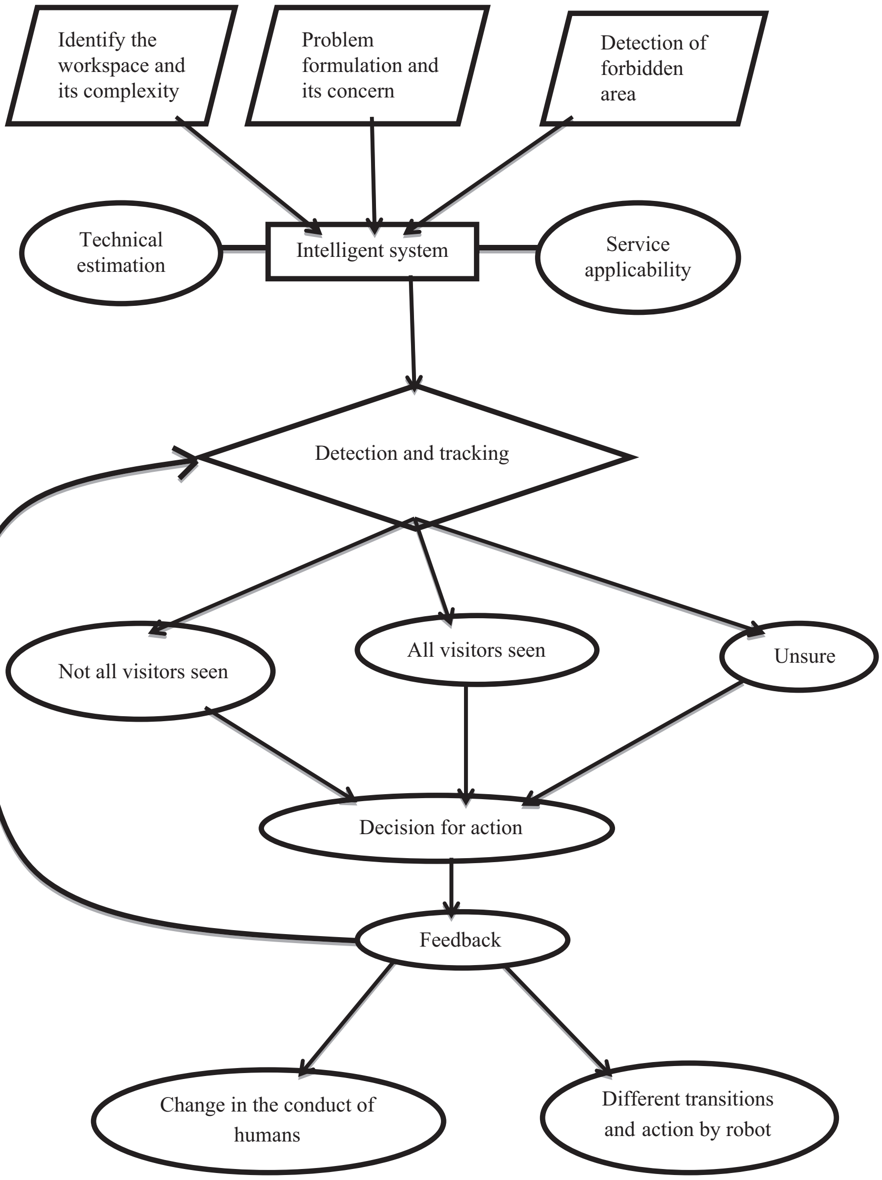

Fig. 1. Architecture of behavior change and robot action plan.

known that the uncertainties will be large due to the visitors being out of field of view. Similarly when the visitors are potentially making a turn, the uncertainties are bound to be large. The robot in such cases anticipates uncertainties and adapts its socialistic response by moving rather than stopping, which is contradictory to the first behavior.

The third novelty of the approach is at the architectural level. A robotics problem is typically solved in a multi-layer hierarchy. A challenge here is that each component also accounts for the socialistic interaction and feedback. The problem is solved as three concurrent behaviors: the reactive navigation mechanism that uses social potential field; the uncertainty-based speed management module that restricts the robot's speed based on specific context and uncertainties; and the higher-level sequencer that operates the overall robotic tour. Both speed manager and sequencer have an audio feedback to the visitors in extreme cases and coordinate so as not to pass simultaneous audio feedback.

Finally, the approach is tested on the Pioneer LX robot at the Robotics and Machine Intelligence Laboratory of the Indian Institute of Information Technology, Allahabad. The robot was made to take small groups on a tour, which it did well. The approach was also tested with a baseline approach. The baseline approach did not make the robot socially react by slowing down and use the contextual

Table I. A summary of related works.

SN	Paper	Description of work
1.	[5]	Target tracking for unstructured crowed scene based on correlated topic model and a collection of behavior distributions
2.	[6]	People tracking using social force, environment restrictions and Kalman filter for the multi-hypothesis tracking
3.	[7]	Collection of hidden Markov models trained on motion patterns to record the temporal and spatial variation in the crowd's motion
4.	[8]	Mathematical concepts and discrete probabilistic velocity obstacle are used to compute the probability of collision and reactive obstacle avoidance
5.	[9]	Motion planning is applied in the dynamic, uncertain and cluttered environment. An optimal feedback control is designed using stochastic dynamic programming. Partially closed-loop receding horizon control strategy is developed
6.	[10]	Kinodynamic constrained robot is planned using rapidly exploring random tree (RRT) which minimizes the potential filed
7.	[11]	Potential field-based method is designed to provide the surety to converge to target using elliptical obstacle functions for repulsions
8.	[12]	Robot helicopter is used for tracking moving objects on the ground. Oxford Aerial Tracking System is modeled as an enhancement to potential field. Artificial neural network is used for trajectory prediction
9.	[13]	A probabilistic model is developed for human detection and motion using laser sensor data using leg pairs
10	[14]	An interactive planning approach is designed using motion prediction based on learning the behavior. This interactive motion planning algorithm is based on velocity obstacle
11.	[15]	Motion pattern of the people is learnt using expectation maximization algorithm and hidden Markov models preserving the belief about the person's position
12.	[16]	A real-time obstacle avoidance approach using artificial potential field using obstacle geometric modeling
13.	[17]	Simulation of bird flocking modeling flow of information in a directed and network topology of the agent changes and events
14.	[18]	Design of a controller and coordinating mechanism for a group of small robots in the formation of a geometric pattern with stability
15.	[19]	Navigation strategy for multiple mobile robots modeled as particles. Nearest neighbor, multi-neighbor and mixed-nearest neighbor tracking methods are used
16.	[20]	Vision-based navigation landmark recognition system using genetic algorithm for pattern recognition and region of interest identification. Can update the data repository with new patterns
17.	[21]	Artificial landmark system is used to complete the task of self-localization and navigation using Mobile Robot codes (objects and invariants features)
18.	[22]	A distributed and modular software architecture of the tour guide robot integrating mapping, localization and planning. Different Markov model-based filters are used
19.	[23]	Humanoid robot adapts its speed with respect to the human walking based on side distance, moving distance and parallel distance

information. The robot simply stopped after it could no longer track a visitor. The proposed approach was socially better and resulted in tours completing earlier due to absence of false positives.

2. Related Work

This section lists some of the most related works from the literature. A summary is also provided in Table I. Robots are increasingly getting common in regular human life. As the interaction increases between the humans and the robot, it is very important to develop a safe and reliable navigation of the robot amidst humans. Generally, robots work in a dynamic environment, so the robots must be able to navigate amidst people with their accurate goal position. In dynamic environments, the robot needs the ability to detect and sense the position of the humans in the workspace so that the navigation can be performed in a complicated terrain without collisions. Detection and sensing of moving humans[5–7] have been examined in the area of computer vision and robotics, and these approaches depend on *a priori* motion model which is typically based on globally optimized parameters for every trajectory during the plenary sequel data. Generally, navigation in a noisy, complex and dynamic ambience is stimulated by prediction of different forms of paths. A probabilistic velocity obstacle approach[8] has been combined to the dynamic occupancy grid for the robot navigation with the assumption that the obstacles have a constant linear velocity. In ref. 9, a robot planning framework was designed which provided the human's anticipated future location, and this information also reduced the suspicion of predicted belief state. Another work described in refs. 10 and 11 used potential-based methodology. For instance, in refs. 12–15, the navigation performance was improved, and high-fidelity human motion model was developed that controlled predictive uncertainty.

In the robot guidance system, the potential function methodology provides an elegant paradigm to specify multiple constrains and goal.[16] In general, reactive navigation methodology can also be developed by coupling attraction from goal with repulsion to obstacles. There are a lot of applications of potential field in motion planning, navigation and multi-robot coordination. Global behavior of the robot and human can be examined through the superposition of various potential functions,[16] such as motion to goal behavior, obstacle avoidance. Here, the evolution of behavior depends on the position of nearest robots.

A recent utilization of artificial behavior was simulation of a school of fish and flocks of bird in the field of computer graphics. The experimental results in this field have been obtained from the work of Reynolds.[17] A simple egocentric model was developed for the flocking which was instantiated in every associated member of simulated set of birds. The work[17] involved formation of complete and close group behavior, although individual agents could only sense their local surrounding and very close neighbors. Multi-robot behaviors have also achieved recent awareness due to their stability and dynamics.[18,19] The work and experiments were based on the analysis of group dynamics and their stability; however, obstacle avoidance cannot be done by this approach. Moreover, behavior formations are completely integrated with the navigation behavior and obstacle avoidance. In general, navigation of a mobile robot can also be applied in various applications of service robotics. So, in order to employ the robot in crowed areas, it should be able to detect the social behavior of the human and also respond to them. To utilize this system, different techniques and methodologies of computer vision can also be used.

There exists related works that use the signals in the workspace to enhance and support the navigation scheme in robotics. Here some methods use reference points or specific marks to initially localize the robot in the workspace; although in some other methods, an artificial vision approach has been applied to visualize them. In ref. 20, the approach used a visual perception and recognizer system for the indoor workspace which had the utility to check the identity at the office. To enhance the utility of the mobile robot, a topological navigation strategy[21] was developed. This topological navigation approach was also applied at the indoor workspace where an artificial mark was used and this artificial mark was called as the Mobile Robot (MR) code. In addition, a topological map was also required in this approach. It stored very less amount of information. This was major drawback of this method and also its dependency on the lighting situation.

Nowadays, it is anticipated from robots to communicate and interact with surrounding people. Hence, robots need some basic tools in order to socially coexist with humans. In general, robot navigation is a very mature, broad and exploring area of robotic. A large number of endeavors exist in the robot navigation which elaborate that the robots are capable of navigating in challenging and difficult environments.[22] Although, a large number of social interactive methodologies are needed.

Numerous methods have been produced to empower the mobile robots to guide the humans in a particular way. Few researchers have started to study how a robot might adjust its velocity when moving besides a human, in environments like library.[23] Fidelity and security are the major factors for the successful establishment of the mobile robots in a human abode area. In various literatures, security and safety of the robot are confirmed by preventing the people from approaching the mobile robots.

The approaches in the literature largely assume a reasonable knowledge of the human position and delve into the design of socialistic behaviors as a response to the human. These are also adapted to specific applications. However, the robots when operating in such environments are subjected to uncertainties due to limited field of view, vision uncertainties, need of sudden turn, unanticipated motion of the human, etc. While the literatures develop better vision models to mitigate such uncertainties, it is also important to develop socialistic model to adapt the robot behavior to act under uncertainty and to convey the same to the human. Such adaptations of behavior have not been attempted in the literature. Since the response is specific to the context and application, the paper takes the specific application of a robot operating as a tour guide.

3. Design Philosophy

In this work, we design the best strategy with the following principles: (i) while the robot is navigating all the visitors in a group and if any visitor leaves the group, then the robot should not continue the tour and must stop and wait for the visitor to join the group, (ii) the robot may need to make sharp turns with visitors following. The robot should tackle the sharp turns itself and allow the visitors to tackle the same in a manner of their liking. The robot should not wait for the visitors to be exactly at the rear of the robot at every instant of time. (iii) The false positives should be as small as possible or the robot must never wait even if all visitors are approximately around. (iv) If the visitors are detected to be missing, the robot must attempt regrouping by making an announcement. (v) Only one robot is available for navigating the visitors.

In the robotic tour, only one robot and visitors are taking place. So, the robot is the captain of the visitor group. The captain robot finds a path and moves for its several visit sites along with the visitors. Since we have used only one robot, this robot is also a captain to watch the visitor's activity behind it. If any visitor leaves the group, then the robot de-accelerates until its velocity becomes zero and thereafter it makes an announcement for the visitors to join a group and waits for them to assemble. After the visitors join the group, the robot moves forward for its next task. Robot has to move in the formation to complete its tour with its estimated position. In this task, the robot also reaches at corners which require it to make a sharp turn to get to the goal. The robot should make such turns in which time the visitors may not be visible due to the limited field of view of the camera.

In the approach, the robot has to do the following tasks: (i) the robot acts as a captain and navigates visitors in a group, (ii) the robot moves in the forward direction on the tour path and looks backward direction to know whether the visitors are following it or not, (iii) the robot also makes an announcement if it reaches a visit site to tell about the place and also makes an announcement when visitors leave the group or missing visitors rejoin the group, (iv) the robot is capable to increase or decrease its speed depending on the visibility of the visitors. Doing so, the robot must complete the entire tour of guiding the visitors in a group.

3.1. Navigation strategy modeling

Consider the robot's pose in the real world as q_R, the rear camera at the robot with a pose q_C and a visitor's (specifically the visitor's face) position as q_V. The camera is assumed to be at the center of the robot, with an additional height, looking at the rear, which gives the transformation between the robot and the camera (T_R^C) as Eq. (1):

$$T_R^C = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & h_C \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Here, h_C is the height of the camera. The -1 denotes that the camera is looking backward or a Z axis rotation of π. The transformation between the visitor and the robot ($T_V^R(q_V, q_R)$) can be obtained with a knowledge of their global positions. The visitor as seen in the image of the camera is given by Eq. (2):

$$q_I = \zeta T_R^C T_V^R q_v \tag{2}$$

Here ζ is the distortion-corrected calibration matrix. However, since the camera has a limited field of view, the visitor will be visible in the image only when the visitor is within the field of view, given by the function $f_V(\zeta T_R^C T_V^R q_v)$, which takes the visitor's projection into the camera as an input and checks for bounds.

Ideally, the visitor is supposed to follow the robot whose generic control law is given by Eq. (3):

$$\begin{bmatrix} v_V \\ \omega_V \end{bmatrix} = \begin{bmatrix} K_1\,(d\,(q_v, q_R) - r) \\ K_2\,(atan2\,(y_R - y_V, x_R - x_V) - \theta_V) \end{bmatrix} \tag{3}$$

Here K_1 and K_2 are the gain constants. $d(q_V, q_R)$ is the distance between the robot and the visitor. r is the sum of radii of the robot and visitor and a little extra safety distance, θ_V is the orientation of the visitor, (x_R, y_R) is the position of the robot and (x_V, y_V) is the position of the visitor.

Under the ideal circumstances, if the robot is stationary, the visitor approaches and stops at the safety distance. If the robot moves, the visitor adapts and moves. The visitor will be able to catch the robot as long as the robot moves slower than the allowable speed of the visitor, which is true with most indoor mobile robots which are limited by speed. The motion needs to be corrected for obstacles. However, the visitor is a social being and will therefore not adhere to the ideal control laws. The visitor can get attracted by other things temporarily, leave for reasons like a phone call, etc. In such a case, a well-known convention is that the robot reacts to the person. One possible mechanism is to have an attractive force between the robot and the visitor, which may make the robot follow the person on the deviated track, a behavior which is inadmissible for the application.

The other mechanism is to make the robot wait when the visitor has temporarily left the group and resume the tour when the visitor rejoins. However, the challenge is to correctly identify the gesture of the person. A few steps away from the robot could be due to an interesting landmark of interest or the person avoiding an obstacle, when the robot should not wait; while it could also be due to the visitor leaving to buy snacks, when the robot should wait. In this paper, we consider the problem when the person is out of bounds of the camera and cannot be tracked, or the bounds of the view function $f_V(\zeta T_R^C T_V^R q_v)$ do not hold. The view function depends on both the pose of the visitor and the robot that discussed separately in the next two subsections.

3.2. *Slowing down of the robot*

This subsection is devoted to the out-of-view conditions $f_V(\zeta T_R^C T_V^R q_v)$ due to the visitor alone. In general, the robot navigates the visitors if all the visitors follow the robot. Conventionally, if any visitor does not follow the robot, the robot stops which however produces a jerk. Moreover, the person may be present and not within the camera's field of view. If the robot suddenly stops, the person does not know about the reason and the person shall never come within the view. We solve this issue in this navigation strategy so that robot does not produce any jerk. Here, we have designed the slowing behavior of the robot. The robot navigates with its normal speed and suddenly if any visitor is not detected, then the robot does not stop but reduces its speed until its speed becomes zero, when it stops. Consequently, if the missing visitor is detected again by the robot, then again robot the increases its speed to normal. Due to this slowly guide behavior, the robot suddenly does not stop and also false positives are handled. Furthermore, since the robot is moving, it is possible that the missing visitor reappears to the robot and the presence of the visitor is reascertained, since the camera and its orientation change with the robot. The robot acceleration or deceleration depends on the availability of the visitors behind the robot.

Within the limited and uncertain sensing, the decision to wait and affirm that the visitor has temporarily left the group is when it is not seen for a threshold amount of time (t_{stop}), or Eq. (4):

$$left_{\mathrm{V}} = true \text{ iff} f_V(\zeta T_R^C T_V^R q_v(t)) = false \ \forall t - t_{stop} \le t \le T \tag{4}$$

Here T is the current time. The probability that the visitor has temporarily left the group is given by Eq. (5), which has a value of 1 when Eq. (4) holds and slowly increases from 0 to 1.

$$P(left_V) = \min\left(\frac{T - t_{lastSeenV}}{t_{stop}}, 1\right) \tag{5}$$
$$t_{lastSeenV} = \arg\min_{\mathrm{v}} \ \arg\max_{f_V(\zeta T_R^C T_V^R q_v(t)) = true} t$$

The minimization over all visitors is to make a decision based on the visitor which is not seen for the long time. Instead of making a brisk decision of stopping when it is ascertained that the visitor is missing, the behavior is smoothened by allowing the robot to slow down as its confidence of the visitor to have left increases. The maximum permissible speed v_R^{pref} is hence given by Eq. (6):

$$v_R^{pref} = \mathrm{v}^{\max} P(left_V) = v^{max} \max\left(\frac{T - t_{lastSeenV}}{t_{stop}}, 0\right) \tag{6}$$

Here v^{max} is the maximum allowable operating speed of the robot without considering the uncertainty constraints.

3.3. Handling sharp robot turns and corners

The other case when the view function $f_V\left(\zeta T_R^C T_V^R q_v\right)$ does not hold is due to the motion of the robot. While the robot typically operates at small speed causing a little change in position, the turn of the robot or change in orientation can make the visitor outside the field of view. As the robot moves, it may have to take sharp turns to avoid obstacles or tackle corners in path. In such cases, the visitors will get out of view. Every after making the turn, the visitors may not be in the view as the visitors are themselves making a similar turn, following the robot. In such cases, the robot should not slow or stop. Slowing or stopping of the robot will create a false positive as the visitors are still there. The stopping of the robot will imply that the visitors need to quickly come behind the robot, which may not be possible as there may be no space in the corner. In this navigation strategy, we have solved this issue. In such a case, the robot must not account for the visitors because visitors will not be able to follow the robot while the robot is making a sharp turn. Although at every state except corner, robot is looking behind it to know the visitors' appearance.

The problem is again to detect such a situation and to take corrective actions in the navigation strategy of the robot. It may initially appear that angular velocity is a good measure of sharp turns. However, numerous control mechanisms may produce fluctuating control signals, which can be hard to interpret for such decision-making. Hence, the angle to sub-goal is used as a measure to detect sharp turns and corners.

The path of the robot to goal or visit sites may be broken into multiple sub-goals. As soon as the robot reaches near a sub-goal, the next sub-goal in the path to goal acts as the current sub-goal. This continues till the last sub-goal (always the goal) is the current sub-goal. This is as per the working of a fusion of deliberative and reactive navigation mechanism.

Let s be the current sub-goal in pursuit of visit site g, which is the ith sub-goal in the deliberative path from source to visit site g. A typical motion of the robot is to go travel in a straight line from the current position to the sub-goal, in which case the robot nearly always faces the sub-goal. Any large deviation from the sub-goal means that the robot had to severely disorient itself to avoid an obstacle. Alternatively, it means that the current sub-goal changed and the new sub-goal requires a sudden change in the orientation of the robot, denoting a corner. The detection of a sharp turn or corner is hence given by Eq. (7):

$$\begin{aligned} turn = true\ iff \cos\left(\theta_e\right) < \eta \\ \theta_e = atan2\left(y_s - y_R, x_s - x_R\right) - \theta_R \end{aligned} \tag{7}$$

Here, (x_s, y_s) denotes the coordinates of the sub-goal and (x_R, y_R) denotes the coordinates of the robot. η is a constant.

If the robot is found to be turning, no slowing or stopping constraints are applied and the robot continues to move until the turning conditions cease to hold. After the turning conditions cease to hold, the robot is said to have accounted for the turn and the tracking continues. It may be inquisitive to note if the tracking further needs to be off considering even after the robot accomplishes the turn, the visitors will need time to accomplish the same turn, during which time they will not be visible to the robot. This will be naturally handled by the robot as per the slowing conditions.

4. Algorithm Design

4.1. Overall architecture

The overall architecture has three distinct modules which work and share information with each other as shown in Fig. 2. The first module is the speed manager that manages the maximum speed of the the robot and also provides the ability to robot for its acceleration and deceleration. If visitors are continuously detected by the robot, it will move with maximum velocity, while if any person is not detected, the velocity of robot decreases. As the visitors again appear, the velocity increases. Robot will only stop when visitors are not detected for certain time. This module also handles the turning cases, when the robot does not react to missing visitors on making a sharp turn. This module also

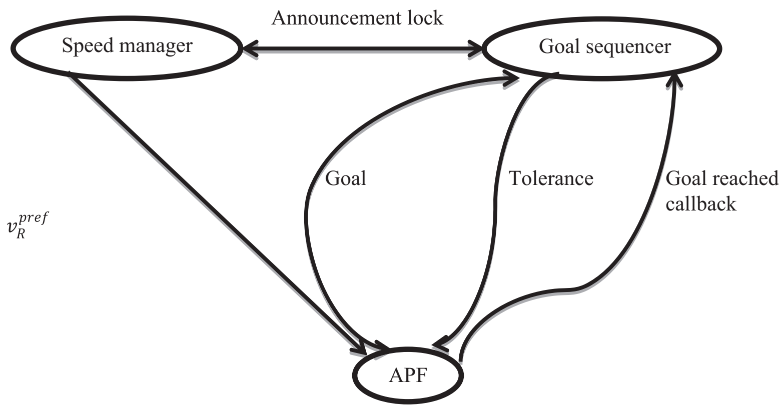

Fig. 2. Goal sequencer.

sends an audio feedback to the visitors, telling them if a visitor is missing or if the missing visitor is now seen and the group must continue.

The second module is the goal sequencer. Here, the different visit sites or goal of the environment (workspace map) are defined. The module sequences different mission sites. Reaching each mission site is further a motion to goal behavior, which is broken into multiple sub-goals. On reaching each mission site, the robot makes the announcements explaining about the site as a tour guide.

An announcement lock system is also built in this scheme. Both the speed manager module and the sequencer module run in parallel and use the speaker as a resource for giving an audio feedback to the visitors for different reasons. It is possible for both audio messages to be invoked together. A module using the speaker must first lock it and unlock it after the announcement. This locking status is shared and communicated between modules. So, when one module uses the speaker for announcement, the other module will not be able to announce anything and will have to wait for the other module to finish the announcement execution.

The APF is the third module which acts as a controller for the robot. Here, potential function and force are applied on the robot, which is used to calculate the control signal for the robot. The APF gets its goal from the sequencer and permissible speed limits from the speed manager. The module also keeps a track of the robot position and when it reaches the next sub-goal (or goal) within the set tolerance limit, it gives a feedback to the sequencer.

4.2. Speed manager

The speed management module sets an allowable navigation speed for the robot, and the module is modeled as a BFSSM for the robot navigation. The term social is deliberatively added to the term to denote that this module interacts with the humans, both in tracking them and using the same information for decision-making, as well as to give them a feedback using audio signals. The module empowers the robot to make the decision of its speed based on the different states as shown in Fig. 3.

The robot is initially at the moving stage, wherein it moves as per the sequencer and controller. In the BFSSM, a timer is assumed to available at every stage. Ideally, the robot keeps moving as per the sequencer and controller. However, if any visitors are not available for a certain time then the robot adjusts its speed as per the time when the visitors were not seen. Eventually, after t_{stop} amount of time, if the visitors are still not seen, the robot will stop. This gives a soft visual feedback to the group about something is wrong and if the missing visitor is around, it would come back, in which case the state is again changed to moving. On the contrary, if this soft feedback does not help, after t_{announce} time, an audio feedback is provided to get the attention of the missing visitors. The announcement system first needs to be locked so that the other modules are not making an announcement at the same time, which results in an interim waiting for announcement state. As all the missing visitors appear and join their group followed by detection by the robot, then the robot again makes an announcement that all visitors are detected and after the announcement is over, the robot again restarts its journey and navigates all the visitors in a group. The detection of visitors does not happen if the robot is encountering a turn. This cycle continuously repeats till the tour is not completed.

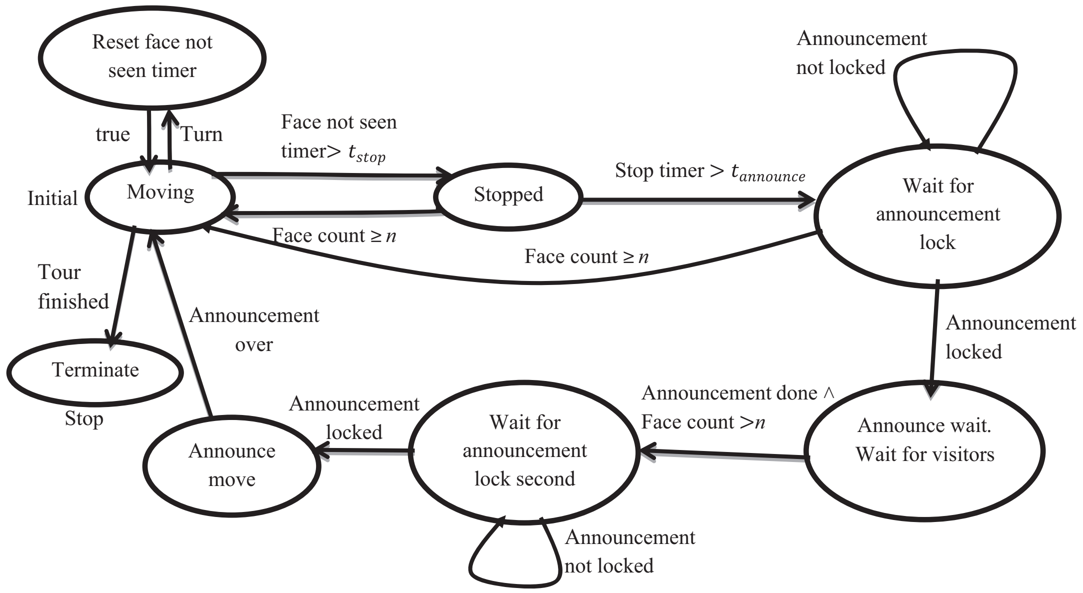

Fig. 3. Task-based B-FSM (Behavior Finite State Machine).

Compressing some of the states of the BFSSM, the simplified algorithm for the velocity manager is given as Algorithm 1. The algorithm receives a *visitorStatus* from the face detection algorithm discussed in Section 4.5. The status is true if all the visitors are visible and false otherwise. Even though all modules interact using Robot Operating System (ROS), the passing of information from face detection to the velocity manager is not done by ROS due to the latency incurred by ROS.

4.3. Goal sequencer

The sequencer is responsible for ensuring that the robot goes from one visit site to the other and completes the entire tour. At the highest level, the problem involves reaching each visit site, explaining the facts about the visit site and continuing to go to the next visit site. Let $V = [v_i]$ be the ordered sequence of sites with $v_i.q$ denoting the location of the site v_i and $v_i.t$ denoting the information to be read about site v_i. After all visit sites have been witnessed, the robot reads out the concluding lines for the tour. The problem is hence to sequence all the visit sites and narrate facts about them in a sequence.

Reaching every visit site is a classic motion planning problem. The reactive solution to motion planning is known to get the robot stuck at local optima. The choice of method for this approach is using a reactive low-level controller so that the robot can circumvent the numerous dynamic moving people in the area. The problem is hence seen as a fusion of deliberative and reactive planning technique. The deliberative planning is used to get a deliberative path from every visit site to the next visit site. The deliberative path is in the form of a sequence of sub-goals or $v_i.\tau_d = [s_{i,j}]$, where $v_i.\tau_d$ denotes the deliberative trajectory to reach the visit site v_i. The robot reaches in pursuit of a sub-goal. As soon as the robot reaches within the tolerance limits of the sub-goal, the sub-goal is said to have reached and the next sub-goal (or goal) is loaded. If the robot reaches the goal, the announcement system plays the informational material about the visit site or goal. The overall algorithm is given by Algorithms 2 and 3. Here ε is the accepted tolerance for the goal and E is the accepted tolerance for the sub-goal.

4.4. Artificial potential field

The robot seeks the sub-goals using the APF. The robot is equipped with a laser, and the laser readings give the information about the obstacles, including the distance from the obstacle and its direction. The repulsive force due to the laser at an angle is φ reporting a distance to obstacle as d_φ, with the robot itself at an angle θ_R is given by Eq. (8). Here α is the repulsive constant.

$$F_{rep} = -\sum_{\phi=\phi_{min}}^{\phi_{max}} \alpha \frac{1}{d_\phi^2} [\cos(\theta_R + \phi)\ \sin(\theta_R + \phi)]^T \tag{8}$$

Algorithm 1 Velocity manager

1) Get visitorsStatus from visitorsChannel
2) Get goalStatus from goalChannel
3) Localize Robot
4) state = "Moving"
5) While (goalStatus≠"end")
 i) Get visitorsStatus from visitorsChannel
 ii) Get goalStatus from goalChannel
 iii) T = Current Time
 iv) If (state ="moving")
  (a) If (visitorsStatus $\vee$ $\cos(\theta_e) < \eta$), $t_{lastSeenV} = T$

$$v_R^{pref} = v^{max} \max\left(\frac{T - t_{lastSeenV}}{t_{stop}}, 0\right)$$

  (b) If (state="moving" $\wedge$ $v_R^{pref} = 0$)
   1. $t_{\mathrm{stopped}} = \mathrm{T}$
   2. state="stopped"
 (v) Else if (state = "Stopped" $\wedge$ $v_R^{pref} = 0$ $\wedge$ T-$\mathrm{t_{stopped}} > \mathrm{t_{announce}}$)
  (a) State = "Wait for announcement lock"
 vi) Else if (state = "Wait for announcement lock" $\wedge$ $\neg$ announcementLock)
  (a) announcementLock = True
  (b) announcementChannel.publish(announcementLock)
  (c) play(wait message)
  (d) announcementLock = False
  (e) announcementChannel.publish(announcementLock)
  (f) State = "wait for visitors"
 vii) Else if (State = "wait for visitors" $\wedge$ visitorsStatus)
  (a) State = "Wait for announcement lock second"
 viii) Else if (state = "Wait for announcement lock second" $\wedge$ $\neg$ announcementLock)
  (a) announcementLock = True
  (b) announcementChannel.publish(announcementLock)
  (c) play(moving message)
  (d) announcementLock = False
  (e) announcementChannel.publish(announcementLock)
  (f) State = "moving"
 ix) speedChannel.publish$\left(v_R^{pref}\right)$

Algorithm 2 Goal sequencer

1) Initialize V
2) i = 1 (goal), j=1 (sub-goal)
3) Publish $\mathrm{s_{ij}}$ as goal
4) If j = len($v_{\mathrm{i}}.\tau_{\mathrm{d}}$), publish small tolerance (goal), else publish large tolerance (sub-goal)
5) while (state not end of tour), sleep

Similarly, the attraction from the sub-goal $s(x_s, y_s)$ with the robot at $q_R(x_R, y_R)$ is given by Eq. (9):

$$F_{att} = \beta d(q_R, s)[\cos(\theta_s)\ \sin(\theta_s)]^T \qquad (9)$$
$$\theta_s = atan2(y_R - y_s, x_R - x_s)$$

The total force is given by Eq. (10):

$$F_{total} = F_{att} + F_{rep} \qquad (10)$$

Algorithm 3 Goal reached status (status)

(The function is called whenever some data is published on the goal reached status)

1) if result = "success"
- **i)** j ← j+1
- **ii)** if(j>len($v_i.\tau_d$))
  - **(a)** while (announcementLock), sleep for a brief period
  - **(b)** announcementLock = True
  - **(c)** announcementChannel.publish(announcementLock)
  - **(d)** Play(v_i.t)
  - **(e)** announcementLock = False
  - **(f)** announcementChannel. publish (announcementLock)
  - **(g)** i←i+1, j←0
    - **1.)** if (i>len(V)), goalChannel.publish("end")
    - **2.)** else goalChannel.publish(s_{ij})
    - **3.)** if(i<len(V) $\wedge$ j $=$ len($v_i.\tau_d$)), toleranceChannel.publish(ε)
    - **4.)** else toleranceChannel.publish(E)
- **iii)** else goalChannel.publish(s_{ij})

The linear and angular velocities of the robot are computed by the same force. Care is given to limit the linear speed to the value set by the speed management module.

4.5. Face detection approach

In this paper, a convolution neural network (CNN) is used to detect the visitor's face. As the visitors appear at the front of the camera (mounted at rear end of the robot), they are visualized by their faces. The network is trained to detect multiple people, so as the visitors increase, the face of the all the visitors are detected. Consequently, a bounding box is also putted on their faces and simultaneously we count the number of faces in the frames.

The framework continuously used for the face detection and localization is taken from ref. 24. The framework is divided into three phases. Figure 4 shows the CNN with Proposal Network (P-Net), Refinement Network (R-Net), and Output Network (O-Net) which is used for the face detection. Here *Conv* and *MP* are the abbreviations for the convolution and max pooling, respectively. Moreover, 1 and 2 are the step size for convolution and pooling.

Assume an image has been given. To design an image pyramid, it is resized into a different scale and this will be the input of the three-sample phase framework. In Phase I, the P-Net is the method of CNN. It is used to produce a candidate facial window and its bounding box with the regression vector. The candidate facial windows are calibrated according to bounding box regression vector. Then, to integrate extremely overlapped candidates, non-maximum suppression (NMS) has been indulged. In Phase II, every candidate window is filled into a R-Net of the CNN. Huge number of false candidates are eliminated by the R-Net. A calibration is performed with the bounding box regression and NMS is regulated. Phase III is similar to Phase I. The major aim of this step is to identify the complete area covered by the human's face with very high ministration. Here, five facial landmarks are the output of the O-Net. Several researches have been carried out for face detection. CNN with a very few filters[25] with less diversity have some restrictions. Here, statistics of the filters are minimized from 5×5 to 3×3 to eliminate the computation time, although depth is increased to achieve a good performance. In this method, a better performance is obtained with a minimum running time. Here after the convolution, a nonlinear activation function Parametric Rectified Linear Unit[26] is used which is the new extension of Rectified Linear Unit.[27]

The first subproblem in the pipeline is to detect faces and non-faces. This is also called as problem of two-class taxonomy. Cross-entropy loss has been applied for each sample c_i. It is written in the mathematical form as Eq. (11):

$$M_i^{det} = -\left(n_i^{det}\log(P_i) + \left(1 - n_i^{det}\right)(1 - \log(P_i))\right) \tag{11}$$

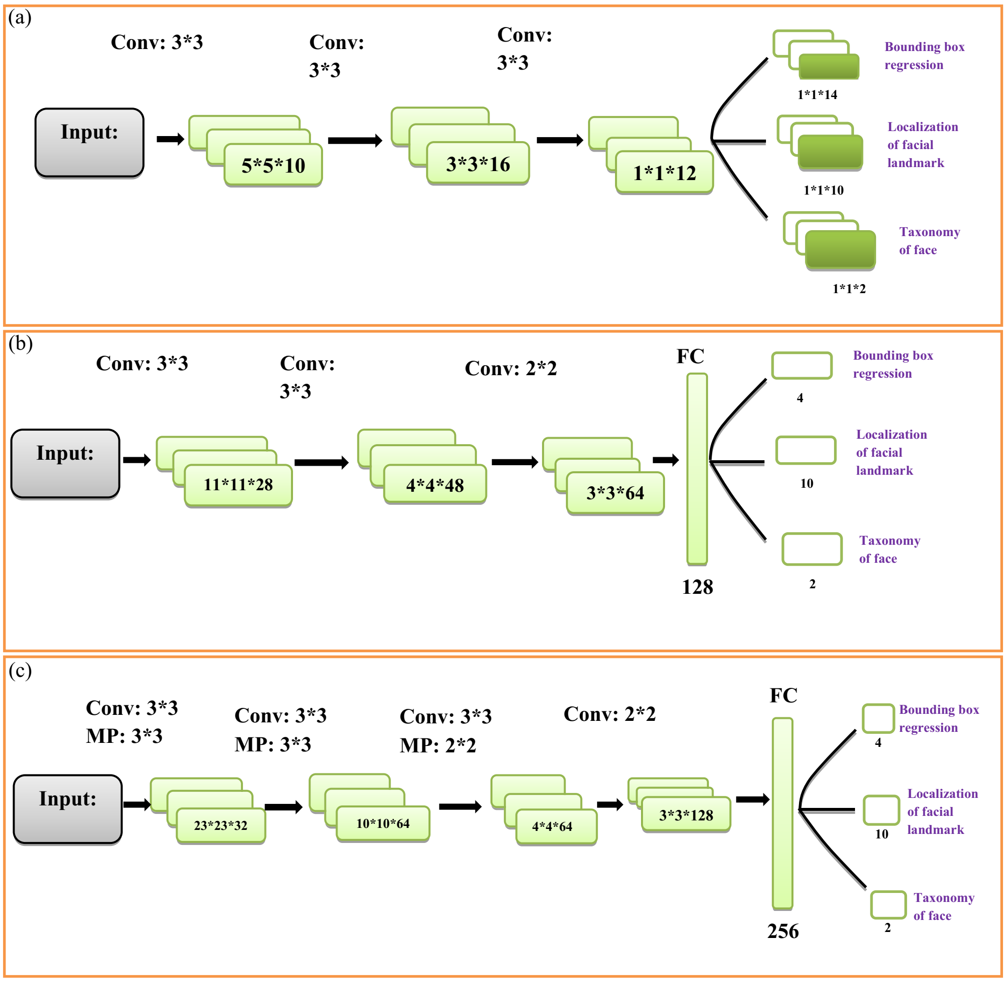

Fig. 4. (a) Sample sketch of Proposal Network (convolution neural network). (b) Sample sketch of Refinement Network (convolution neural network). (c) Sample sketch of Output Network (convolution neural network).

Here, n_i^{det} is representing a ground truth label and P_i is the network-obtained probability and also it also implies that sample P_i being face.

The next subproblem is to draw a bounding box. Offset is estimated for every candidate, and its closest ground truth (width, height, top and left of each bounding box) is used for training. The aim of this method is also called as problem of regression. Furthermore, Euclidean loss is indulged for each sample c_i. It is represented in Eq. (12):

$$M_i^{box} = \|\hat{n}_i^{box} - n_i^{box}\|_2^2 \tag{12}$$

Here, regression target is represented by $\hat{n}_i^{box}$ which is obtained from the network, and n_i^{box} is the special symbol used to represent the ground truth coordinate which includes four coordinates (width, height, top and left), so it implies that $n_i^{box} \in R^4$.

The last subproblem is to detect the facial landmarks. Euclidean loss is minimized in this procedure. So it is also a problem of regression as similar to the bounding box regression. It is mathematically expressed in Eq. (13):

$$M_i^{landmark} = \|\hat{n}_i^{landmark} - n_i^{landmark}\|_2^2 \tag{13}$$

In Eq. (13), $\hat{n}_i^{landmark}$ represents the facial landmark coordinates which are computed from the network and $n_i^{landmark}$ denotes the ground truth coordinate for i^{th} sample. In addition, nose, right mouth corner, left mouth corner, right eye and left eye are five facial landmarks, so it implies that $n_i^{landmark} \in R^{10}$.

In each CNN, multiple processes and steps are involved and performed. So in the learning process, distinct type of the training images are utilized, such as face image, non-face image and image with

Algorithm 4 Face detection algorithm

1. Initialize P-Net, R-Net, and O-Net
2. Import and initialize the camera
3. While(tour is operating)
   a. I ← Image from camera
   b. F ← Detect the face by the convolution neural network which is P-Net, R-Net, and O-Net on I with minimum size $[\eta_x^{min}, \eta_y^{min}]$
   c. F ← Select faces strictly within the image
   d. if (length(F) = noVisitors ∧ (width(f) = η_x^{max} ∧ height(f) = η_y^{max}) ∀ f ∈ F
      i. visitorsChannel.publish(true)
   e. else
      i. visitorsChannel.publish(false)
4. Display image I and faces F

moderately aligned face. For instance, if a sample belongs to the background, then only M_i^{det} is evaluated for remaining another two losses that has been set to zero. This is modeled as a sample-type indicator. Furthermore, the target of this learning procedure is represented in Eq. (14):

$$min \sum_{i=1}^{N'} \sum_{j\in\{det,box,landmark\}} \aleph_j \beth_i^j M_i^j \tag{14}$$

Here, the symbol N' is the number of training samples and $\aleph_j$ is used to represent task importance. Often, the values $\aleph_{landmark} = 0.5$, $\aleph_{det} = 1$ and $\aleph_{box} = 0.5$ are fixed for P-Net and R-Net, but for the O-Net these values are fixed as $\aleph_{landmark} = 1$, $\aleph_{det} = 1$ and $\aleph_{box} = 0.5$ to ensure the very good localization of the facial landmarks. $\beth_i^j$ is used to denote the sample-type indicator and $\beth_i^j \in \{0, 1\}$. To train CNN, a stochastic gradient decent is used. In general, original classifier is trained followed by involving traditional hard sample mining. Online hard sample mining is applied for the taxonomy of face and non-face. Specifically, in the each mini-batch, losses are sorted which are found in the forward propagation from every samples, and the hard samples are chosen among the topmost loss samples. These hard samples are used to evaluate the gradients only at the back propagation. So as a result, easy samples are avoided since they are very less likely to strengthen the detector in training phase.

There are few limitations of the face detection approach. (i) The camera should be upright at nearly the human height. The performance reduces when the camera looks at the person at an angle. (ii) The methodology can detect people far away; however, to get real-time results, too small face detections from people far away are neglected for computational speedup. (iii) The technique cannot detect faces when the human misorients himself/herself and is looking away from the camera. (iv) The robot needs to have a high computing ability to do all the calculations in real time.

The face detection algorithm[24] uses a multi-task network in order to simultaneously detect the faces of multiple visitors. Visitors too far away from the robot are virtually non-existing as they exhibit a nonsocial behavior. Therefore, bounds on size $[\eta_x^{min}, \eta_y^{min}]$ are applied, and any visitor's image that does not obey the bounds is called as missing. The face detection algorithm is shown as Algorithm 4.

5. Implementation details

This methodology has been implemented on master ROS framework. Webcam is used to visualize humans and detect their faces. The proposed navigation strategy was tested on simulator first and then on the real robot. We have used Pioneer LX robot to check the proposed methodology. The Robotics and Machine Intelligence Laboratory at IIIT, Allahabad, Prayagraj, was the workspace for navigation. Webcam has tightly fixed on a monopod and this monopod was mechanically fixed on the Pioneer LX robot. ROS ARIA library provided the basic functionalities of operating the robot.

The first problem was to make a map of the laboratory. In order to do this, the robot was first operated in the laboratory. The map is made based on the lidar readings and the wheel encoders of the robot. Thereafter, the different sites of interest were manually marked at the map. These constitute the visit sites that the robot should visit in the given order. The ordering was manually done. The

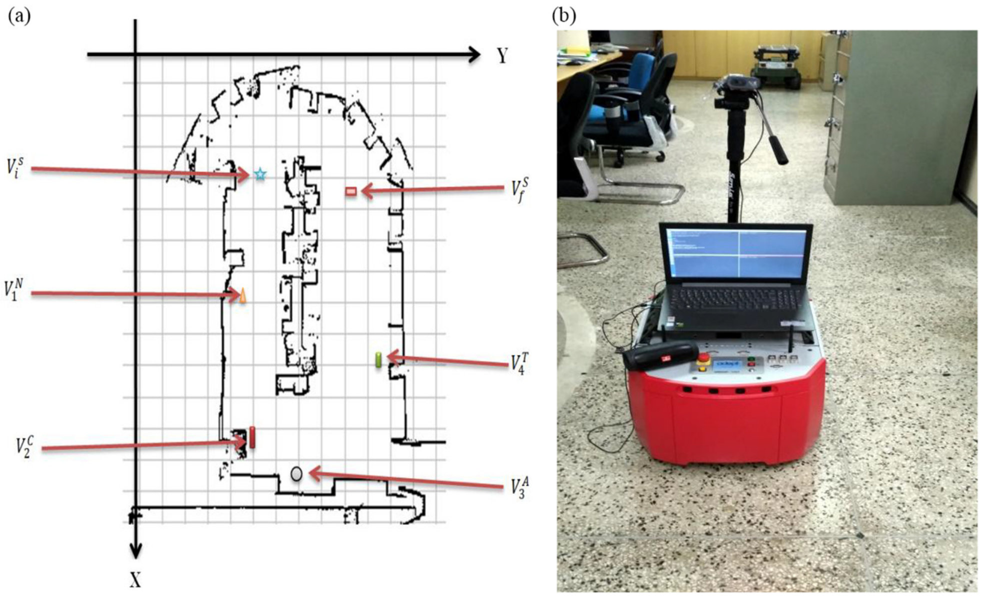

Fig. 5. (a) Different visititing sites on the workspace. (b) Robot setup for visitors navigation.

deliberative path between the visit sites was also manually specified. In most cases, a straight line path was possible, not requiring the need of sub-goals.

The algorithm was executed in two systems. The first system was a laptop with system configuration 8 GB RAM and Core i5 processor. This system did most of the processing including image processing, visitor detection, velocity management and potential field. The other system was the embedded system at the robot which used the commands for the navigation of the robot. Both the systems were connected using the ROS interface.

In Fig. 5(a), our workspace is shown with various visiting sites, while Fig. 5(b) represents the robot setup through a robot master slave. Descriptions of various visiting sites are as follows:

1. V_i^S : This is initial state of the robot or starting point of the robot for visitors' navigation.
2. V_1^N : It is first visit site where *NaO* and *AmigoBot* robots are placed.
3. V_2^C : An in-house developed autonomous self-driving mini car called Autonomous Robot Testbed
4. V_3^A : *Adaptive Modular Active Leg (AMAL)*, an in-house developed prosthetic leg available in the laboratory
5. V_4^T :A social robot for interaction with children named as *Tinku*
6. V_f^S : Last visiting point or final state of the robot where a socialistic mobile robot called SMART is located.

6. Experimental Evaluation and Result

In this section, we discuss about the results of our methodology. This method has been tested on three scenarios. In the first scenario, the robot starts its journey from the starting point and visits the visiting sites and finally reaches the last vising point. In this case, each person follows the robot and no person misses the journey in the middle. The second scenario is the same as the first scenario but both visitors miss the group at two places of the tour. Moreover, in the third scenario, one visitor misses the tour at one place and both visitors miss the tour at another state of the tour. The missing of visitors may be due to cases like receiving a phone call, going off group as the visitor sees something interesting, catching children, etc. While any visitor misses the group or tour, the robot decreases its speed. Meanwhile, if any visitor rejoins the group, then the robot increases its speed and moves toward their goal. If the visitor does not reappear, then the robot stops and announces for the visitor to join the group followed by waiting for the person to appear or join the group. As soon as the person joins the group, robot sends an acknowledgment to the visitors through a voice and continues the tour. All results are also given in the supplementary video.

Table II. Comparison between the proposed method and baseline.

Scenario	Scenario description	Time duration of tour (s)		Time duration of tour excluding visitors not available (s)		False positives	
		Proposed	Baseline	Proposed	Baseline	Proposed	Baseline
1	No visitors leave	**124**	152	**124**	152	**0**	2
2	Two visitors leave at V_1^N and two visitors leave at V_4^T	**154**	205	**119**	172	**0**	3
3	One visitor leaves at V_1^N and two visitors leave at V_4^T	**147**	228	**119**	196	**0**	4

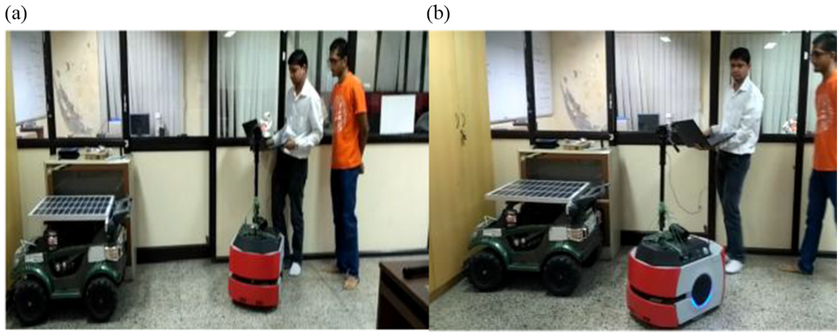

Fig. 6. Robot location at a corner.

The proposed approach uses interesting heuristics and uncertainty-based navigation behavior of the robot. In the literature, there was no competent approach for the same problem that could be used for comparisons. Hence, the comparison of the approach is done on a baseline. The baseline works on the principle of making the robot move as long as all visitors in the group can be tracked. If the visitors can no longer be tracked, the robot stops and announces for the missing visitors. Table II describes the metric comparisons of both the proposed approach and the baseline approach. Three metrics are considered. The first metric is the total time to complete the tour. The robot in both cases does not move when the visitors are missing. Ideally, the robot should only wait for the time when the visitors are not available. Any additional time waiting hence is the robot waiting as the visitor is not currently visible or the robot making a wrong decision. The metric is modified to account for the time when the person was actually not present as adjudged by a human looking at the scenario. The false positives, or cases when the robot believes the person is not present and makes an announcement, while the person is actually there, however may not be visible to the robot, is another metric. The proposed approach easily outperforms the baseline on all metrics by a good factor. The robot finishes its tour comparatively early with the proposed approach and moreover the false positives are also zero, while in the baseline method the false positives exist which delays the tour. This should be visualized as the visitors waiting and adjusting themselves to be within the view of the robot which can be a painful experience often observed with the adoption of technology in nonideal circumstances.

In the baseline, some of the times the visitors are present at the rear end of the robot; however, the robot is not able to detect them and stop. When the robot reaches at a corner and makes a sharp turn toward its next visit site, then the robot is not able to detect the visitors and stops and waits for the visitors to appear in front of the camera, while there is not enough space at the rear end of the robot as shown in Fig. 6(a). Waiting makes the problem complicated as the visitors need to occupy the little space at the back of the robot to even make the robot move. In the proposed approach, we resolved this issue and the robot can easily tackle such turns as shown in Fig. 6(b), which is a novelty and contributes toward a better performance.

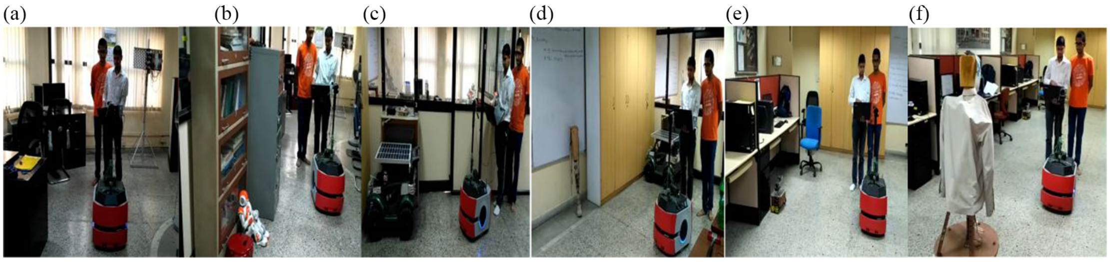

Fig. 7. Robotic tour for scenario 1. (a) Robot at its initial state and is ready to tart its tour. (b) Robot successfully arrived at its first visiting site where NaO and AmigoBot are placed. (c) Robot at its third visiting site where autonoumous car is parked. (d) Robot succeefully made a sharp turn and reached at third state, AMAL. (e) Robot at the fourth vising site where Tinku a socialistic robot is present. Meanwhile, the robot also took a sharp turn to reach here. (f) Robot reached at its last vising site and SMART robot is located here. Also the robot terminates here.

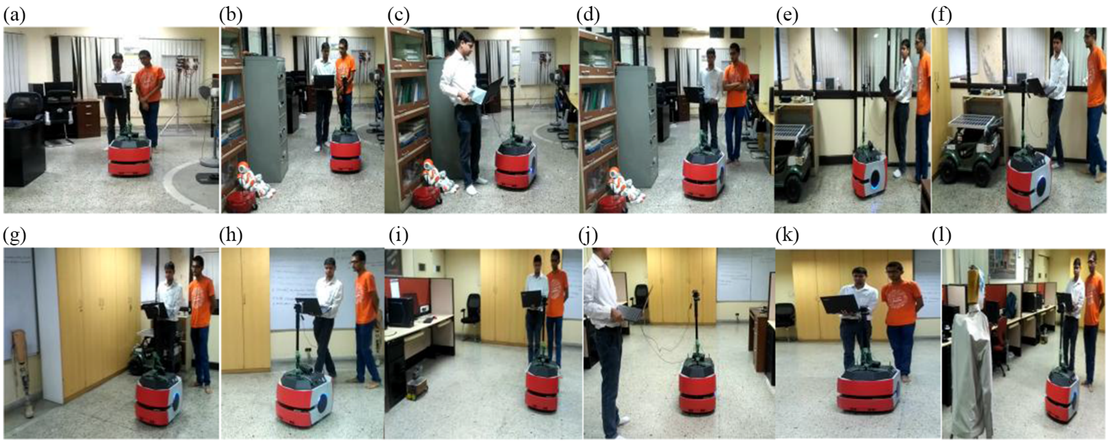

Fig. 8. Robotic tour. (a) Robot at its initial state and is ready to start its tour. (b) Robot successfully arrived at its first visiting site where NaO and AmihgoBot are placed. (c) Here, both visitors left the group and not detected by the robot. So, robot waited for those visitors to join the group. (d) All visitors are detected behind the robot. (e) Robot at its third visiting site where the autonoumous car is parked. (f) Robot succeefully made a sharp turn to proceed to its next visiting site. (g) Robot successfully reached at the third state, AMAL. (h) Robot again made a sharp turn at the corner to proceed its next visiting point. (i) Robot at fourth vising site where Tinku a socialistic robot is present, meanwhile robot. (j) All the visitors are not detected by the robot behind it so the robot waited for the visitors. (k) Visitors appeared behind the robot and were detected by the robot. (l) Robot reached at its last vising point and SMART robot was located here. Also, the robot terminates here.

The result of our first test case for the first scenario is shown in Fig. 7. Here, every visitor followed the robot and also false positives were zero and the tour completion time was small. In this scenario, the robot started its journey at its initial state and completed its journey at its last visiting site. The robot could also tackle sharp turns easily. Different snapshots from the tour are shown in Fig. 7.

Further, in the second test case tour, the robot navigated all the visitors and also made sharp turns at the corner very easily. In this scenario, visitors were also following the robot somewhere in the middle, so the robot decelerated and stopped, and made an announcement to wait for them. As the visitors joined the group and appeared behind the robot, the robot has started its tour. All these cases are shown in Fig. 8.

The third test case was similar to the second one. The only difference was that at one state only one visitor left the group. So, the robot waited for that visitor. Some specific snapshots of this tour are shown in Fig. 9.

Similarly, the tour is discussed for one case for the baseline. False positives existed here and the robot stopped while the visitors were present behind the robot and were following the robot. The major problem arose when the robot reached at the corner and could not move as the visitors were not visible. While it was making the turn with this method, then it suddenly took 90° rotation at the corner. In these circumstances, the visitors could not follow the robot. Some of the time while robot

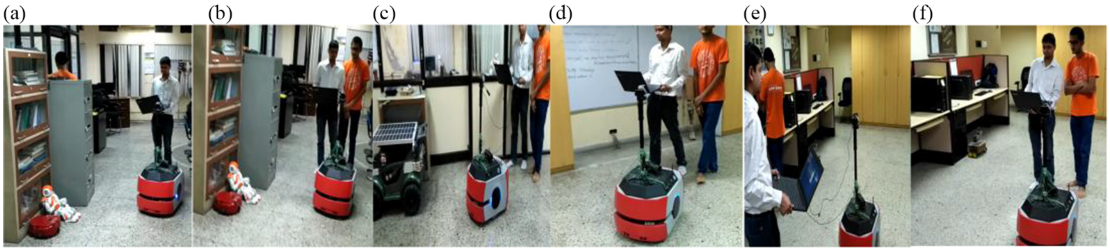

Fig. 9. Robotic tour. (a) Here, only one visitor left the group while the other visitor was available, still the robot waited. (b) All the visitors are detected by the robot and it resumed its journey. (c) Robot successfully made a sharp turn at the corner to proceed to its next visit site. (d) Robot again made a sharp turn. (e) All the visitors moved away in different directions and were not seen by the robot, so the robot waited for them. (f) All the visitors come near the robot in the rear side and are detected by the robot, so the robot moved for its next visit site.

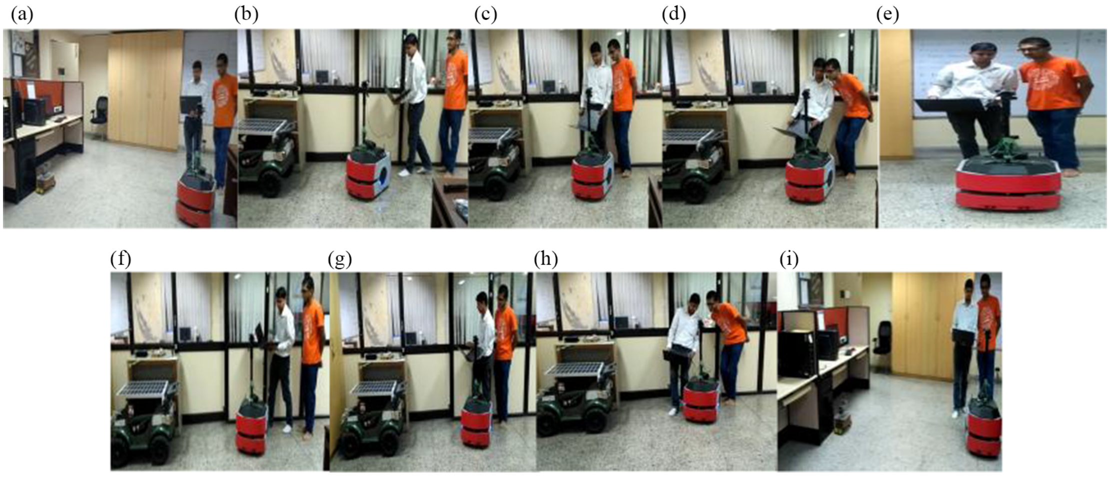

Fig. 10. Robotic tour for the baseline. (a) Robot suddenly took a 90° rotation and produced false positives while both visitors were present behind the robot. (b) Robot took a turn at a corner in such a way there is not enough space behind the robot. (c) Visitors tried to appear behind the robot; however, the robot did not detect them. (d) Visitors changed their face orientation along the camera and their face could not be detected by the robot. (e) Robot took a turn with 90° rotation and generated false positives. (f) When robot was taking a turn, there was no sufficient space behind the robot and that produced false positives. (g) Visitors tried to appear behind the robot. (h) Visitors appeared behind the robot but they were not detected by the robot, so they changed their face orientation according to view of the camera mounted on the robot. (i) Robot took a 90° rotation to reach its next visit site and also produced false positive although visitors were available behind the robot.

was rotating at the corner, then there was not enough space behind the robot where visitors could place themselves. Even when the visitors were trying to place, their face was not easily detected, and the view of the camera mounted on the robot had to be changed manually to look behind. These situations are shown in Fig. 10.

At different time duration of the robot navigation, angle to goal (rotation, given by the cosine value) and velocity of the robot have been determined for each case. The graphical representations of these trajectory, angle, and velocity of scenario 1, scenario 2 and scenario 3 are shown in Figs. 11–13 respectively. All the plots in blue color are for the proposed approach and the red color is for the baseline.

With these metrics of comparison between the proposed approach and the baseline, we state that the proposed method takes lesser time for the tour completion as compared to the baseline. Moreover, the speed of the robot is disproportional to the time duration of missing visitors. If the visitors are available at the front of the camera mounted on the robot, then the speed of the robot will be maximum defined by the speed controller, and if visitors are not present behind the robot then speed of robot will decrease till zero. So, these are distinct parameters to evaluate the performance of our proposed approach work and we found that this proposed method has the best performance in terms

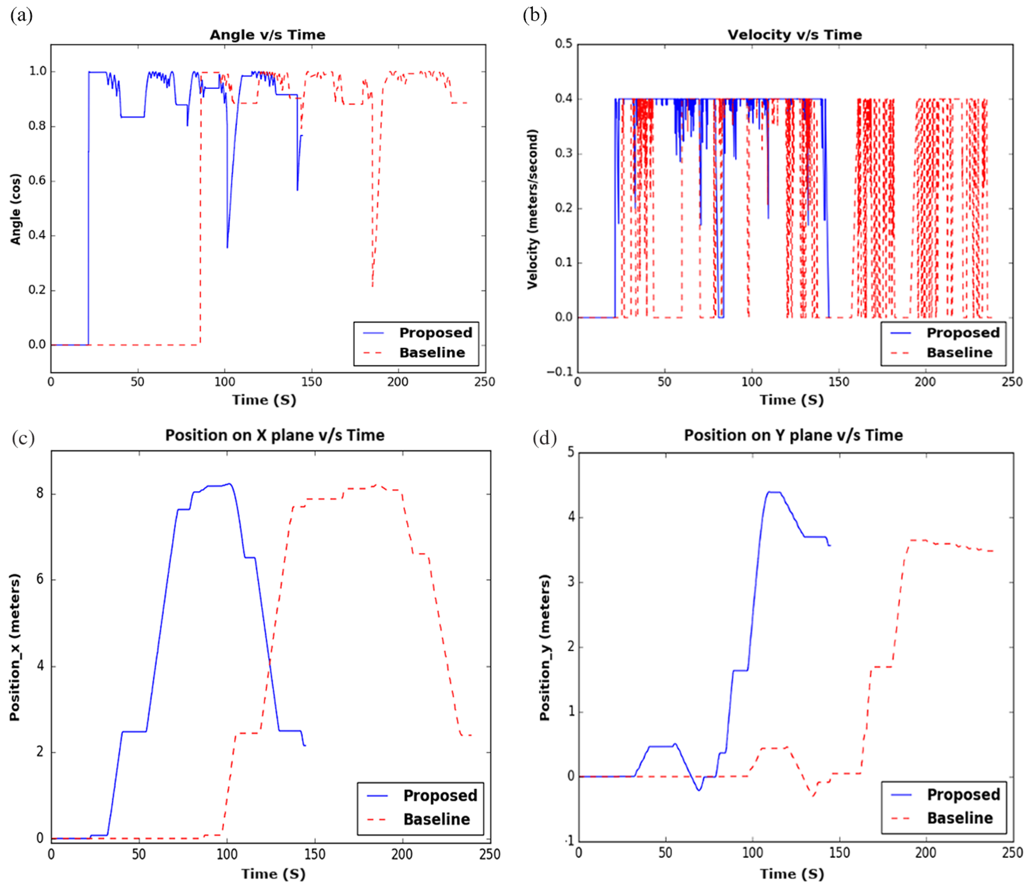

Fig. 11. Test case I comparison between proposed method and baseline.

of journey completion time, with the robot slowly decreasing and increasing velocity as its behavior. The robot does not get stuck at a corner while taking a turn and also maintains a sufficient space so that the visitors can easily follow the robot and get detected by the robot.

It may be noted that the proposed scheme is for small group tours only. The face detection algorithm has a large scalability and will continue to have a reasonable response time and is not a bottleneck. However, in large groups, it is impossible for all the visitors to come inside the same image. In tours, visitors typically get close to the guide and in case the group is large, the visitors spread all around. The limited field-of-view camera can thus only see a subset of visitors. This is an actual problem with tour guides in large groups as well, wherein counting the visitors only takes place at major stops by actually roaming around the group before departing.

7. Conclusion

In this paper, a novel robot navigation strategy based on BFSSM was presented. The robot operating in social context with interaction of the people is a complex problem as the people seldom act erroneously that is typically not modeled in the robotic behavior. The aim of the paper was to take a specific example of robot operating as a tour guide and to adapt the conventional robot behavior to account for the missing visitors. Further, cases, wherein the visitors are not within the field of view of the camera for some time and the case where the robot makes a sharp turn and hence the visitors are not visible, were handled. The software architecture with all such exceptional cases could have been overly complex. The problem was hence broken into modules and the modules could interact with each other using messages.

The validation of this strategy was demonstrated on distinct scenarios of simulator and several real-life experiments on the Pioneer LX robot. In contrast to a baseline method, the proposed strategy

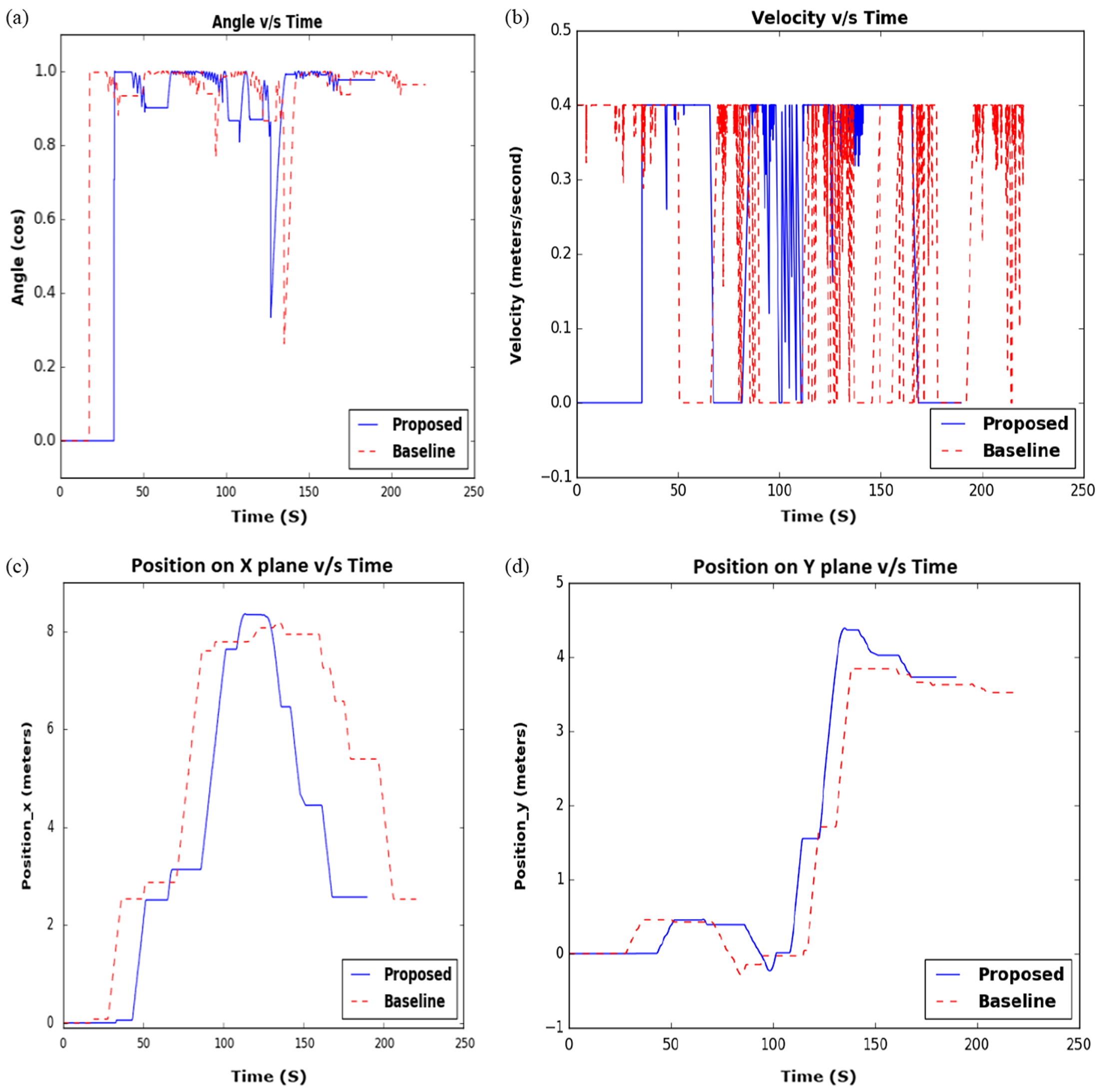

Fig. 12. Test case II comparison between proposed method and baseline.

handled realistic conditions like when robot reaches at a corner and takes a sharp turn for the next goal. In the proposed methodology, the false positives are reduced to zero and the robot does not stop suddenly, while it decelerates and reduces its speed very slowly form the current speed to a zero speed. So, the robot does not produce a jerk while stopping. The velocity of the robot is kept inversely proportional to the time duration of missing visitors. Therefore, this work can be applied in the certain real-life robotics area, such as navigating the visitors in museums, shopping malls. The proposed approach results in a more socially acceptable and a faster tour as compared to a baseline approach.

In the future, we will apply and integrate a strategy to select the largest clearance as an enhancement of the approach. So, even if multiple paths exit at a sharp turn, then the robot will choose the path which has a large space or gap between the boundaries of the workspace. In this scheme, the possibility of a failure to reach the goal and the robot getting close to the boundary will get resolved. Furthermore, the behavior of the robot will be enhanced by studying more socialistic behaviors between tour guides and people and transferring the same into the robots. The approach needs to be extended to large tours in a multi-robotic setting.

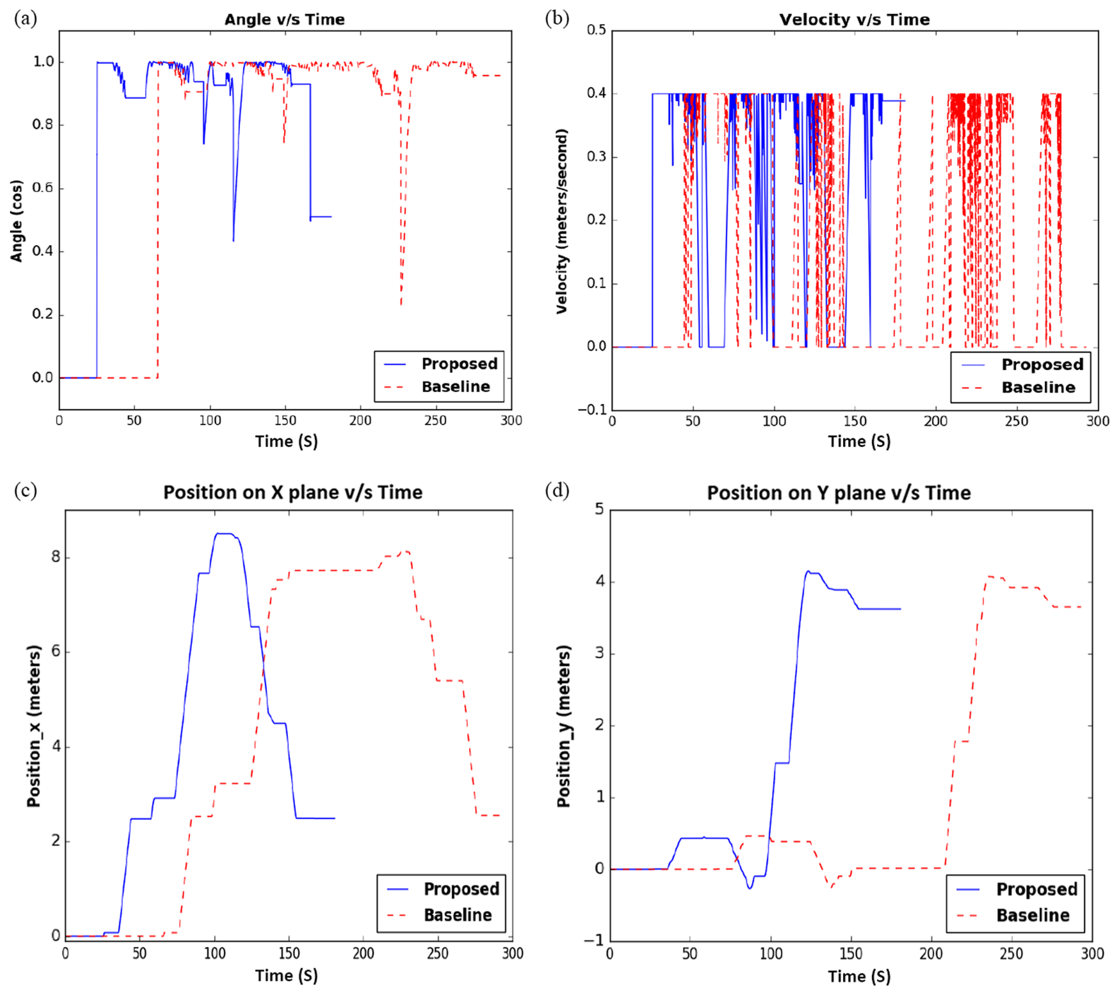

Fig. 13. Test case III comparison between proposed method and baseline.

Acknowledgment

The work is supported by the Science and Engineering Research Board, Department of Science and Technology, Government of India through grant number ECR/2015/000406 and the Indian Institute of Information Technology, Allahabad.

Supplementary material

To view supplementary material for this article, please visit https://doi.org/10.1017/S0263574720000259.

References

1. K. Dautenhahn, S. Woods, C. Kaouri, M. L. Walters, K. L. Koay and I. Werry, “What is a robot companion-friend, assistant or butler?.” In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE (2005) pp. 1192–1197.
2. A. Zelinsky, “A mobile robot exploration algorithm.” *IEEE Trans Robot Autom* **8**(6), 707–717 (1992).
3. A. Franchi, L. Freda, G. Oriolo and M. Vendittelli, “A randomized strategy for cooperative robot exploration.” In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE (2007) pp. 768–774.
4. J. H. Reif and H. Wang, “Social potential fields: A distributed behavioral control for autonomous robots.” *Robot Auton Syst* **27**(3), 171–194 (1999).
5. M. Rodriguez, S. Ali and T. Kanade, “Tracking in unstructured crowded scenes.” In *2009 IEEE 12th International Conference on Computer Vision*, IEEE (2009) pp. 1389–1396.

6. M. Luber, J. A. Stork, G. D. Tipaldi and K. O. Arras, "People tracking with human motion predictions from social forces." In *2010 IEEE International Conference on Robotics and Automation*, IEEE (2010) pp. 464–469.
7. L. Kratz and K. Nishino, "Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes." *IEEE Trans Pattern Anal Mach Intell* **34**(5), 987–1002 (2012).
8. C. Fulgenzi, A. Spalanzani and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid." In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE (2007) pp. 1610–1616.
9. N. E. Du Toit and J. W. Burdick, "Robotic motion planning in dynamic, cluttered, uncertain environments." In *2010 IEEE International Conference on Robotics and Automation*, IEEE (2010) pp. 966–973.
10. M. Svenstrup, T. Bak and H. J. Andersen, "Trajectory planning for robots in dynamic human environments." In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE (2010) pp. 4293–4298.
11. N. Pradhan, T. Burg and S. Birchfield, "Robot crowd navigation using predictive position fields in the potential function framework." In *Proceedings of the 2011 American Control Conference*, IEEE (2011) pp. 4628–4633.
12. H. Helble and S. Cameron, "3-d path planning and target trajectory prediction for the oxford aerial tracking system." In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE (2007) pp. 1042–1048.
13. S. Thompson, T. Horiuchi and S. Kagami, "A probabilistic model of human motion and navigation intent for mobile robot path planning." In *2009 4th International Conference on Autonomous Robots and Agents*, IEEE (2009) pp. 663–668.
14. F. Large, D. Vasquez, T. Fraichard and C. Laugier, "Avoiding cars and pedestrians using velocity obstacles and motion prediction." In *IEEE Intelligent Vehicles Symposium*, IEEE (2004) pp. 375–379.
15. M. Bennewitz, W. Burgard, G. Cielniak and S. Thrun, "Learning motion patterns of people for compliant robot motion." *Int J Robot Res* **24**(1), 31–48 (2005).
16. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots." In *Proceedings. 1985 IEEE International Conference on Robotics and Automation, vol. 2*, IEEE (1985) pp. 500–505.
17. C. W. Reynolds, "*Flocks, herds and schools: A distributed behavioral model*." Vol. 21(4), ACM (1987) pp. 25–34.
18. Q. Chen and J. Y. S. Luh, "Coordination and control of a group of small mobile robots." In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, IEEE (1994) pp. 2315–2320.
19. P. K. C. Wang, "Navigation strategies for multiple autonomous mobile robots moving in formation." *J Robot Syst* **8**(2), 177–195 (1991).
20. M. Mata, J. M. Armingol, A. de la Escalera and M. A. Salichs, "A visual landmark recognition system for topological navigation of mobile robots." In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 2, pp. 1124–1129 (IEEE, 2001).
21. F. Wen, K. Yuan, W. Zou, X. Chai and R. Zheng, "Visual navigation of an indoor mobile robot based on a novel artificial landmark system." In *2009 International Conference on Mechatronics and Automation*, IEEE (2009) pp. 3775–3780.
22. W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun, "Experiences with an interactive museum tour-guide robot." *Artif Intell* **114**(1–2), 3–55 (1999).
23. E. Sviestins, N. Mitsunaga, T. Kanda, H. Ishiguro and N. Hagita, "Speed adaptation for a robot walking with a human." In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, ACM (2007) pp. 349–356.
24. K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks." *IEEE Signal Process Lett* **23**(10), 1499–1503 (2016).
25. H. Li, Z. Lin, X. Shen, J. Brandt and G. Hua, "A convolutional neural network cascade for face detection." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2015) pp. 5325–5334.
26. K. He, X. Zhang, S. Ren and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." In *Proceedings of the IEEE International Conference on Computer Vision*, (2015) pp. 1026–1034.
27. A. Krizhevsky, I Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks." In *Advances in Neural Information Processing Systems*, (2012) pp. 1097–1105.