International Journal of Intelligent Robotics and Applications
https://doi.org/10.1007/s41315-020-00125-4

REGULAR PAPER

A human-aware navigation method for social robot based on multi-layer cost map

Fang Fang[1,2] · Manxiang Shi[1,2] · Kun Qian[1,2] · Bo Zhou[1,2] · Yahui Gan[1,2]

Received: 2 December 2019 / Accepted: 18 March 2020

Abstract

Most of the current human-aware navigation methods of service robots focus on improving the reactive navigation of local path planning without considering the global environment. A global path planning method is proposed based on the global scope of pedestrian perception and multi-layer cost-maps. Firstly, personal space and group interaction are modeled as social cost based on pedestrian perception, and then multi-layer dynamic cost maps are generated containing the social cost at different time-steps based on pedestrian trajectory prediction, which provides social constraints for global path planning. Secondly, the global path planner searches for the optimal state with heuristic cost function based on the multi-layer dynamic cost-maps. Considering the huge calculation of heuristic search and the limitation of the length of trajectory prediction duration, the 'plan-prediction-execution' cycle is introduced for the dynamic planning, which improves performance in the dynamic environment. Finally, compared with the traditional path planner in the simulation scenes including pedestrian movements and group interaction, the experimental results show that the path length, the execution time is shorter, and the comfort distance of the person/group is more social in our method. Through the actual scene experiments, the advantages of handling situations of planning timeout and adjusting trajectories dynamically after introducing the 'plan-prediction-execution' cycle are verified, which can meet the comfort and society of human-aware navigation.

Keywords Service robot · Path planning · Multi-layer cost-map · Human-aware navigation

1 Introduction

Human-aware navigation aims to improve social skills of the robot during navigation. Compared with traditional navigation methods, human-aware navigation must satisfy the requirements of comfort and society (Kruse et al. 2013), which require the robot to keep an appropriate distance with people and adopt a suitable approach strategy, and follow social norms, such as avoiding passing through a group of people. Rios-Martinez et al. (2015) reviews the social norms that robots should follow from the perspective of sociology and psychology, and proposes social constraints that robots should follow in terms of personal space, interaction space, and activity space.

A variety of approaches have been proposed for human-aware navigation, which can be divided into two classes:

1) Navigation with the path planner

The path planner provides commands of motion control for the mobile robot based on known or partially known environmental maps, including global path planner and local path planner. Global path planner calculates an optimal collision-free path from the current location to the target location, which can be realized by A* algorithm, RRT (Rapidly exploring Random Tree) (Lavalle and Kuffner 2011) and so on. Local path planner calculates and adjusts the trajectory to deal with obstacles in the environment, which can be implemented by DWA(Dynamic Window Approach) (Fox et al. 1997) and TEB (timed elastic band) (Rösmann and Feiten 2012).

Most human-aware navigation methods integrate social constraints into the path planner to meet the requirements

✉ Fang Fang
ffang@seu.edu.cn

[1] School of Automation, Southeast University, Nanjing 210096, China

[2] Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing 210096, China

Published online: 12 April 2020

of human-aware navigation by pedestrian modelling. Social cost functions are applied to model multiple social constraints. Kriby (2010) and Mateus et al. (2019) employ hybrid asymmetric Gaussian function to represent proxemics for modelling comfort distance with pedestrians. Truong and Ngo (2016, 2017) takes human states (position, orientation, motion and hand poses) and social interaction into account to model extended personal space and social interaction space with defined Gaussian-like function. Pierson et al. (2018) introduces a set of congestion cost function that maps the density and motion of obstacles to an occupancy risk and constructs a planning space from this set. SFM (Social Force Model) is widely employed to model the movement of pedestrians and robots. Ferrer and Sanfeliu (2014) and Yang et al. (2019) extend SFM to describe the interactive force, referred to as social force, with human and environment. Besides, many methods are proposed to predict the trajectory or intention of pedestrians for human-aware navigation, such as predicting meeting points of pedestrians with Bayesian inference (Escobedo and Spalanzani 2014; Ferrer and Sanfeliu 2014) and predicting interaction intention between pedestrians and robots with off-line learning (Park et al. 2016). Bai et al. (2015) utilizes Markov Decision Processes to deal with uncertainty in the predictions of trajectory of pedestrians. Based on pedestrian modelling, the path planner employs path planning algorithm to realize human-aware navigation. Ferrer and Sanfeliu (2014) applies the RRT algorithm and improves local obstacle avoidance based on extended SFM. Hekmati and Gupta (2018) incorporates predicted trajectories of dynamic obstacles in the planning process of DWA algorithm. Khambhaita and Alami (2017) extends TEB local path planner by introducing prediction and optimization of human trajectories, in which the robot uses the same environmental map to coherently plan its own trajectory and predict plausible pedestrians trajectories and provide a solution where both human and robot can move optimally. Gaydashenko (2018) combines D* Lite path planning algorithm and trajectory prediction with Deep Learning techniques for robot navigation through human crowds.

2) Navigation without the path planner

Some researchers propose approaches without the path planner. Kretzschmar et al. (2016) presents a novel approach to model the cooperative navigation behavior of humans in terms of a mixed distribution that captures both the discrete navigation decisions, such as going left or going right, and the natural variance of human trajectories. The model parameters of this distribution are learnt to match with the observed behavior in terms of user-defined features via inverse reinforcement learning. Jiang et al. (2018) proposes a real-time personalized human-following approach in which desired human following parameters are represented by a desired image that is learnt from human-driven demonstrations. Then a non-vector space controller is developed to drive the robot to follow the desired image. Chen and Everett (2017) puts forward an approach based on Deep Reinforcement Learning, which aims to learn an optimal robot policy based on the observed state of the robot and nearby pedestrians. The policy is represented as a neural network which inputs the state information on the robot and pedestrians and outputs the robot velocity.

This paper focuses on the navigation with the path planner. Most of existing approaches usually aim at improving reactive navigation of local path planning, lacking the consideration of the global environment, and treat pedestrians as obstacles, which ignores the social attribute like group activities. To address these issues, the major contribution of this paper is to integrate global scope of pedestrian motion information and group information into a defined dynamic cost map through trajectory prediction, which can reduce the circuitous trajectory because of ignoring the global pedestrian out of the local pedestrian perception, and also develop a global path planning method based on a multi-layer costmap (Lu et al. 2014), which introduces 'plan-prediction-execution' cycle for dynamical path planning to improve the adaptability to the complex and variable environment.

The architecture of our human-aware navigation method is demonstrated in Sect. 2. Section 3 presents the implementation of our method, which includes details of pedestrian modeling, dynamic costmap and global path planner. Section 4 presents the experimental results and analysis. A brief conclusion of this paper is given in Sect. 5.

2 Architecture of our human-aware navigation method

The architecture diagram of human-aware navigation method proposed in this paper is shown in Fig. 1, which is mainly divided into the following three parts:

1. Global Pedestrian Modeling: This module uses multi-view pedestrian tracking via laser and multi-RGB-D cameras to obtain the pose and motion of pedestrians in the global range. Group detection sub-module extracts the motion features for grouping. In addition, a multi-layer social costmap is generated by trajectory prediction, which indicates pedestrian pose and group information in future.
2. Dynamic Costmap: This module obtains global static cost map and a time-varying local obstacle costmap from the map server module, and obtains a multi-layer social costmap from the global pedestrian modeling module. The module will be triggered at the beginning

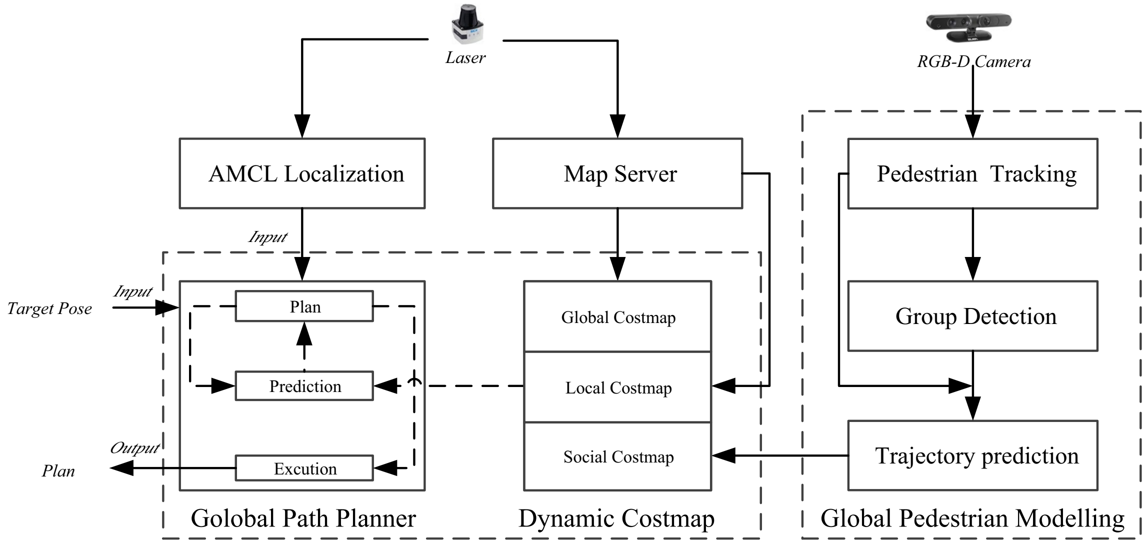


Fig. 1 Architecture of human-aware navigation

of the ‘Plan’ period of the global path planner to provide social constraint in future.

3. Global Path Planner: This module uses the ‘Plan-Predict-Execute’ cycle for dynamic planning. The input is the target pose and the initial pose of the robot obtained through the AMCL Localization module. The output planning sequence is generated for each ‘Plan’ period. At the beginning of each ‘Plan’ period, the trajectory prediction sub-module is triggered and return the dynamic costmap. Then, heuristic search is performed and the current optimal state is returned based on the defined cost function. At the end of ‘Plan’ period, ‘Execution’ period is activated to send the planning sequence to the robot.

3 Implementation of the human-aware navigation method

3.1 Social cost modelling

Based on pedestrian tracking and group detection, personal space and group interaction are combined to model pedestrians in the environment, and integrated into global path planning by introducing social cost.

3.1.1 Personal comfort distance modeling

For every detected pedestrian, a Gaussian cost function $\mathrm{p\_\cos t}_i(x, y)$ is built for personal space (Kirby 2010):

$$\begin{aligned} \mathrm{p\_\cos t}_i(x, y) &= \exp\left(-\left(\left(\frac{d\cos\theta}{\sqrt{2}\sigma_i^x}\right)^2 + \left(\frac{d\sin\theta}{\sqrt{2}\sigma_i^y}\right)^2\right)\right) \\ d &= \sqrt{\left(x - x_i\right)^2 + \left(y - y_i\right)^2} \\ \theta &= a\tan 2\left(\left(y - y_i\right), \left(x - x_i\right)\right) - \theta_i \end{aligned} \tag{1}$$

where x_i, y_i, θ_i is the position and orientation of person p_i and σ_i^x, σ_i^y is the standard deviation of Gaussian cost function, which prohibits the robot from crossing the personal space and satisfies the requirement of comfort in Fig. 3a.

3.1.2 Group interaction modelling

Coherent motion indicator is extracted by pedestrian tracking as the basis of group interaction modeling, including the relative positions Δx_{ij}, the difference of velocity ΔV_{ij} and the difference of orientation $\Delta\theta_{ij}$ between any pedestrian i and pedestrian j, which is found in large social experiments to reflect the group relationships (Moussaïd et al. 2010).

Based on the motion feature of coherent motion indicator, a social relationship graph is constructed as shown in Fig. 2, where the nodes of the graph are tracked pedestrian and each edge of the graph is the social relationship strength R_{ij} between any pedestrian i and pedestrian j among N pedestrians. R_{ij} is the probability output of SVM classifier (Linder and Arras 2014). The SVM classifier inputs the relative positions Δx_{ij}, the difference of velocity ΔV_{ij} and the difference of orientation $\Delta\theta_{ij}$ between any pedestrian i and pedestrian j and outputs whether pedestrian i and pedestrian j belonging to a group. The probability output of SVM means the probability of whether pedestrian i and pedestrian j belong to a group or

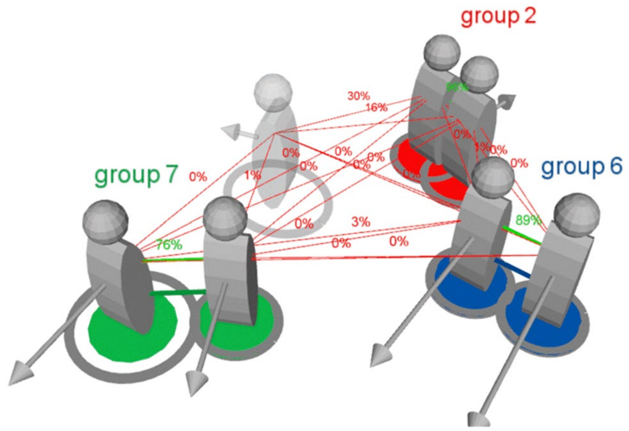


Fig. 2 Group detection

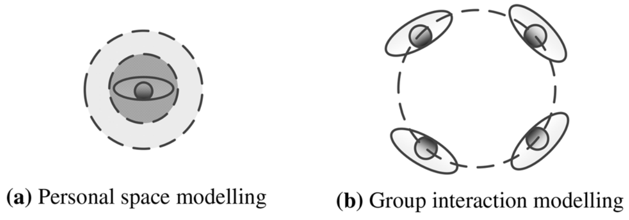


Fig. 3 Social cost

not, which can represent the relationship between pedestrian *i* and pedestrian *j*

$$R = \mathrm{SVM}(\Delta x, \Delta V, \Delta\theta) \tag{2}$$

Then, group detection is completed by pruning and clustering: (1) pruning optimization is performed on the edge of which social relationship strength is weaker than a specific threshold for reducing the computation of clustering. As shown in Fig. 2, the red lines indicate that the social relationship R_{ij} between the pedestrians is very weak, which is usually lower than the threshold value of 0.5, so they are cut off to reduce the computation of the following step of clustering. The green lines indicate the edge after pruning. (2) relationship matrix is generated and groups are detected by clustering methods such as hierarchical clustering.

Circle fitting based on the least-square method for the members of the detected group is performed, after which a Gaussian cost function $\mathrm{g\_cos\,t}_i(x, y)$ is built for every group:

$$\begin{gathered}\mathrm{g\_cos\,t}_i(x, y) = \exp\left(-((\frac{d}{\sqrt{2}\sigma_i^x})^2 + (\frac{d}{\sqrt{2}\sigma_i^y})^2)\right) \\ d = \sqrt{(x - x_i)^2 + (y - y_i)^2}\end{gathered} \tag{3}$$

where x_i, y_i is the position of every group and σ_i^x, σ_i^y is the standard deviation of Gaussian cost function, which prohibits the robot from crossing the group as shown in Fig. 3b.

3.2 Dynamic costmap construction and updation

Dynamic costmap represents the static environment constraints and social constraints over time, which consists of global costmap, local costmap and N-layered social costmap that contains social cost generated by EKF trajectory prediction for N prediction periods, as shown in Fig. 1. The global costmap is constructed by the laser sensor of the robot for the environment while the local costmap is generated by obstacles detected by the laser sensor.The i-th layer dynamic social costmap describes the social cost after i prediction periods. Specially, when the time t exceeds the prediction phase, the corresponding dynamic cost map only contains global costmap and local costmap.

The simulation scenario of the composition of the dynamic costmap is shown in Fig. 4a, where a pedestrian walking straightly in the corridor and a conversation group composed of three persons are detected. Figure 4b represents the global static cost map and local obstacle cost map in the environment where the dotted line represents the expansion area of the obstacle. The social cost of the two continuous prediction periods based on pedestrian trajectory prediction and group detection is shown in Fig. 4c and d, where a right movement between two costmaps indicates the movement of the pedestrian walking straightly, integrating group information and the motion information into the dynamic costmap.

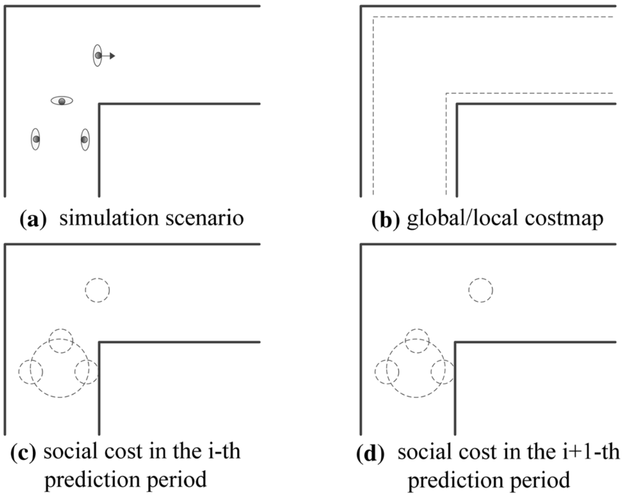

(a) simulation scenario
(b) global/local costmap
(c) social cost in the i-th prediction period
(d) social cost in the i+1-th prediction period

Fig. 4 Dynamic costmaps

3.3 Human-aware global path planning

The global path planning method combines search graph with the robot motion constraint, where dynamic costmap defines the cost function of state switching to evaluate the state in the search graph. Heuristic search is performed in each planning cycle to obtain the best feasible plan sequences.

3.3.1 Problem definition

The state space of robot is discretized by lattice state, which consists of state node and motion primitive. Each state node can change into another state node via motion primitive. Therefore, lattice state transforms the original continuous state space into a search graph, and path planning is converted into an optimal search problem from the initial state to the target state in the graph.

It is assumed that the trajectory that the robot can perform in the interval Δt depends on the current pose of the robot (x, y, ψ), the speed (v, w), the kinematic constraints of the robot and the social cost of the corresponding time t. In order to provide feasible trajectories for differential-driving robot, the state space of the robot is defined as (x, y, ψ, v, w, t). Search graph is generated by discrete states, which can expand to new states in the interval Δt under a limited set of executable operations U

$$U = \left\{ \left(e_x a_x, e_\varphi a_\varphi\right) \middle| e_x, e_\varphi \in \{-1, 0, 1\} \right\} \tag{4}$$

In Eq. (4) a_x, a_φ represent the linear acceleration and the angular acceleration. Each operation generates a motion primitive to start state switching (1) judge whether the motion primitive satisfies the kinematic constraints of the robot: if it is satisfied, the corresponding trajectory of state switching is calculated. Otherwise the operation is abandoned (2) calculate the social cost of the generated trajectory based on the dynamic costmap according to the time of state switching (so the costmap used by each state may be different) (3) accumulate the social cost and path length for the new state.

3.3.2 Search strategy

A* algorithm is employed for solving optimal search problem. The cost function of traditional A* algorithm is

$$f\left(s_i\right) = g\left(s_i\right) + h\left(s_i\right) \tag{5}$$

where $g\left(s_i\right)$ represents actual cost accumulated from initial state s_0 to current state s_i and $h\left(s_i\right)$ represents estimated cost from initial state s_i to target state.

In this paper, the actual cost $g\left(s_i\right)$ is calculated according to the path length and social constraints. The cost of social constraint is equal to the social cost of the state, which contains the cost of personal comfort distance model and the social cost of group modeling designed in Sect. 3.1.

$$\begin{aligned} g\left(s_i\right) &= \omega_p * Path(s_i) + \omega_s * Social(s_i) \\ Path(s_i) &= \sum_{s=s_0}^{s_i} PathLen(s) \\ Social(s_i) &= \sum_{s=s_0}^{s_i} (\mathrm{p\_\cos t + g\_\cos t}) \end{aligned} \tag{6}$$

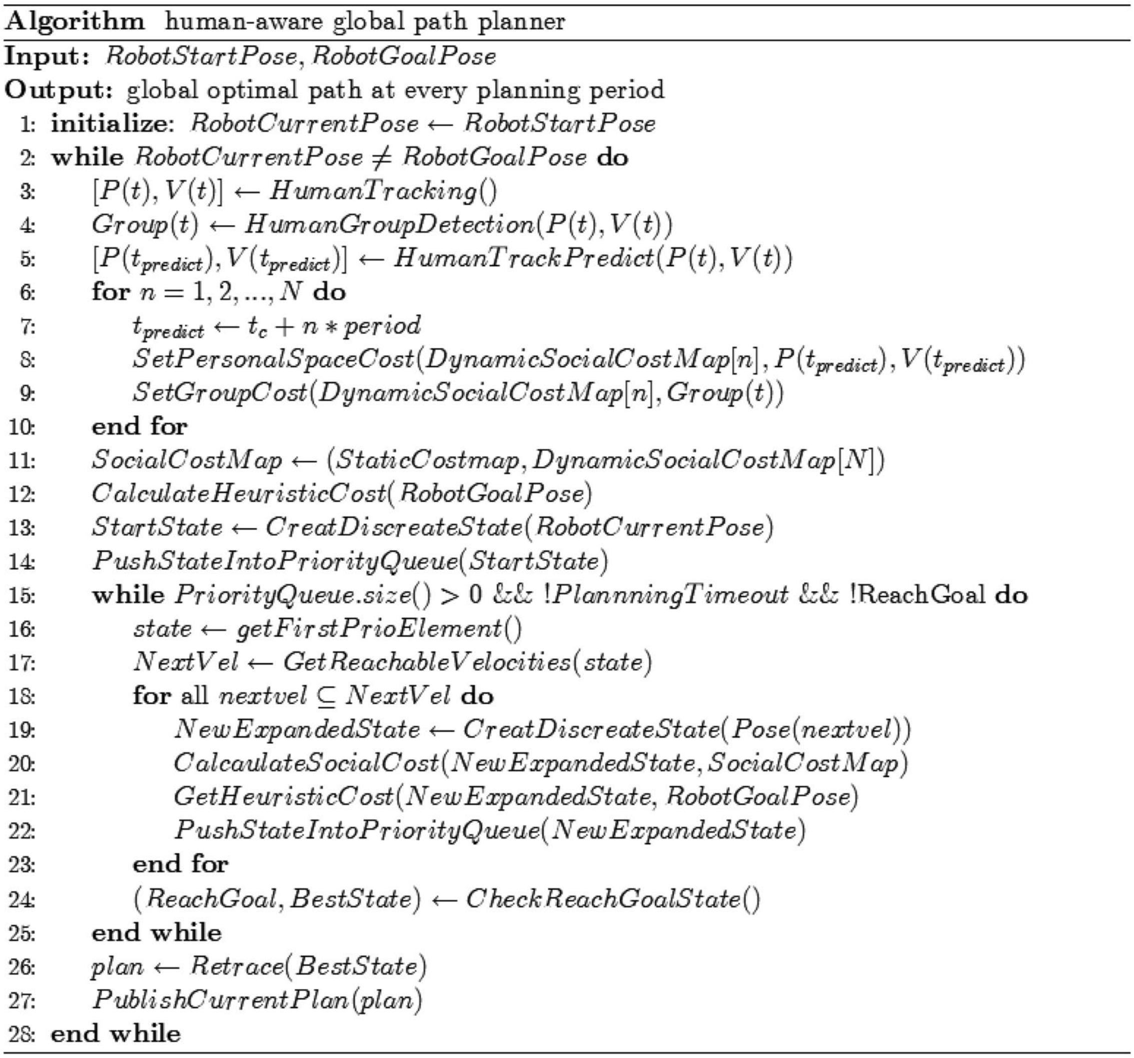
Besides, estimated cost $h\left(s_i\right)$ is proportional to the shortest path to the target pose via Dijkstra algorithm on the static cost ignoring dynamic social cost and motion constraints of the robot.

3.3.3 Global path planning algorithm

Global path planning algorithm is demonstrated in Table 1. The input of the algorithm is the robot's target pose *RobotGoalPose* and the current pose *RobotStartPose*. Before the robot reaches the target pose, the planner divides the time into several equal planning circles for dynamic planning and outputs the current global optimum path *plan* at the end of each planning period. Each planning cycle is divided into three phases for line 3–27 in Table 1.

1. Pedestrian Modelling: After pedestrian tracking module outputs position and speed information of pedestrian $P(t), V(t)$, group detection module outputs the group information $Group(t)$ while the trajectory prediction module outputs the position and speed information of the following N periods $P(t_{predict}), V(t_{predict})$ for line 3–5 in Table 1. Social costmap DynamicSocialCostMap[N] containing social cost of N following periods is updated with global/local costmap *Static*Cost*Map*, which constructs a multi-layer costmap *Social*Cost*Map* for calculating cost of the state in the search graph for line 6–11 in Table 1.
2. Path planning: Line 12 calculates the heuristic cost of global scope according to the target pose. Line 13–14 create the initial state *StartState* of the search graph according to the current pose with calculating cost value according to formula (5) and put it into the priority queue. Line 15–23 continuously get the optimal state in the priority queue and expand the state to generate several new state *NewExpandedState* and calculate the

Table 1 Human-aware global path planning



```
Algorithm human-aware global path planner
Input: $RobotStartPose, RobotGoalPose$
Output: global optimal path at every planning period
1: initialize: $RobotCurrentPose \leftarrow RobotStartPose$
2: while $RobotCurrentPose \neq RobotGoalPose$ do
3:     $[P(t), V(t)] \leftarrow HumanTracking()$
4:     $Group(t) \leftarrow HumanGroupDetection(P(t), V(t))$
5:     $[P(t_{predict}), V(t_{predict})] \leftarrow HumanTrackPredict(P(t), V(t))$
6:     for $n = 1, 2, ..., N$ do
7:         $t_{predict} \leftarrow t_c + n * period$
8:         $SetPersonalSpaceCost(DynamicSocialCostMap[n], P(t_{predict}), V(t_{predict}))$
9:         $SetGroupCost(DynamicSocialCostMap[n], Group(t))$
10:    end for
11:    $SocialCostMap \leftarrow (StaticCostmap, DynamicSocialCostMap[N])$
12:    $CalculateHeuristicCost(RobotGoalPose)$
13:    $StartState \leftarrow CreatDiscreateState(RobotCurrentPose)$
14:    $PushStateIntoPriorityQueue(StartState)$
15:    while $PriorityQueue.size() > 0$ && $!PlannningTimeout$ && !ReachGoal do
16:        $state \leftarrow getFirstPrioElement()$
17:        $NextVel \leftarrow GetReachableVelocities(state)$
18:        for all $nextvel \subseteq NextVel$ do
19:            $NewExpandedState \leftarrow CreatDiscreateState(Pose(nextvel))$
20:            $CalcaulateSocialCost(NewExpandedState, SocialCostMap)$
21:            $GetHeuristicCost(NewExpandedState, RobotGoalPose)$
22:            $PushStateIntoPriorityQueue(NewExpandedState)$
23:        end for
24:        $(ReachGoal, BestState) \leftarrow CheckReachGoalState()$
25:    end while
26:    $plan \leftarrow Retrace(BestState)$
27:    $PublishCurrentPlan(plan)$
28: end while
```

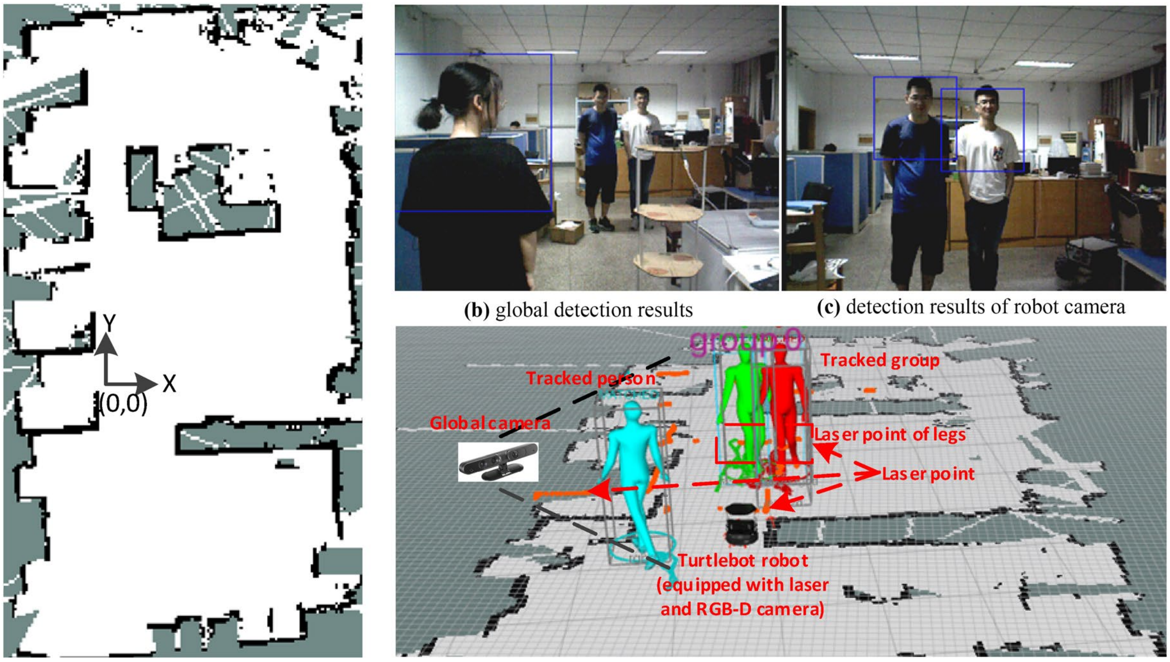
corresponding cost value, then put it into the priority queue. Line 24 checks whether it can reach the target state in the current planning period and returns the best state *BestState* with the lowest cost.

3. Publish plan: The plan sequences of the current planning cycle are obtained by backtracking the best state and are published to the actuator at the end of the 'Plan' period to start a new 'Execution' period for lines 26–27 in Table 1.

4 Experimental results and evaluations

The feasibility of the proposed algorithm is verified by simulation and actual scene experiments with Turtlebot. The simulation scenario experiment uses the planner proposed in this paper (represented by planner A) and the traditional path planner with global path planning using A* algorithm and local path planning using DWA algorithm (represented by planner B) for comparison. The actual scenario experiment mainly verifies the effectiveness of planner A, which introduces the 'Plan-Prediction-Execution' cycle.

Player stage is used as the simulation environment, which can set the same environmental conditions for the planner comparison experiment (including the environment, initial/target pose of the robot, simulated pedestrian etc.). Both the simulation experiment and the actual scene experiment use the Gmapping SLAM method to construct the environmental grip map, as shown in Fig. 5a, and the robot is localized by AMCL algorithm. In the actual scene, multi-view pedestrian detection and tracking work with the laser sensor, RGB-D camera of the robot and global RGB-D camera. Pedestrian detection for RGB-D camera adopts the existing method based on the upper-body template matching, which can overcome changes of body poses and illumination changes. But it is not suitable for large-range detection. The detection



(b) global detection results

(c) detection results of robot camera

(a) environment map

(d) Pedestrian tracking and group detection

Fig. 5 Simulation and actual scene environment

results of RGB-D camera of the robot and global RGB-D are shown in Fig. 5b and c. Pedestrian detection for laser employs existing leg classifier for laser point feature, which works well for large-range detection but has lower accuracy of detection. The laser point features of leg are shown in Fig. 5d. The above pedestrian detection results are transformed into the world coordinate and merged by the nearest neighbor algorithm based on the world coordinate 2D position. Pedestrian tracking performs on data association algorithm, based on which group detection is implemented, as shown in Fig. 5d.

4.1 Simulation experiments

- Scene 1: moving pedestrian in the way of robot

  In this scene, the robot detects a pedestrian moving in the positive x direction at (0, 0) during navigation. Planner A detects the pedestrian through global pedestrian perception at around t = 4 s, and adjusts the global plan in advance in the next planning cycle t = 4.5 s, as shown in Fig. 6a. Planner B can only employ reactive obstacle avoidance through local planning, so the pedestrian is sensed by the laser sensor of robot at t = 8 s. Due to above reasons, the trajectory is adjusted accompanied by large-direction adjustment, as shown in Fig. 6b.

  Simulation experiments are repeated 30 times for two planners. The average execution time, the average path length and the shortest distance between the robot and the person are listed in Table 2. The planner A shortens the execution time and path length by introducing global pedestrian motion information. In addition, due to the comfort distance modeling of pedestrians, the shortest distance between the robot and the person meets the comfort requirement while the planner B can only perform reactive obstacle avoidance. When the robot moves in the same direction with a pedestrian, the local costmap is not cleared in time and the robot is surrounded by the cost generated by the pedestrian, resulting in plan failure, as shown in Fig. 6c.
- Scene 2: group of pedestrians in the way of robot

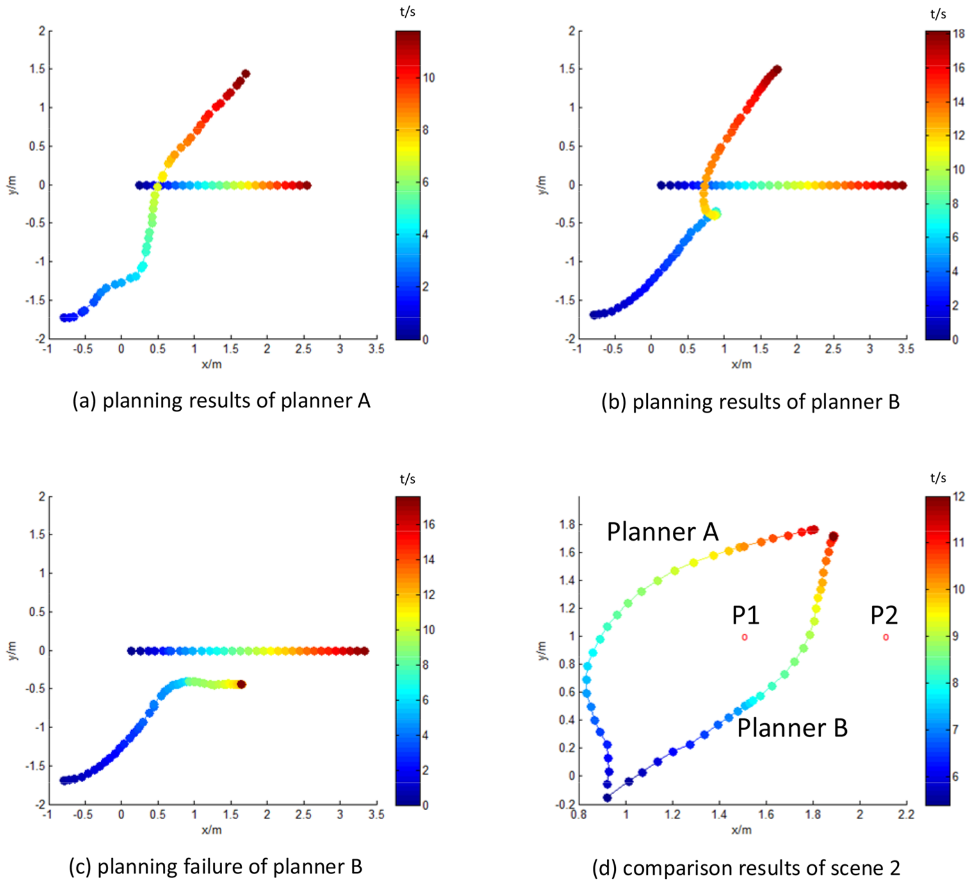
  In this scene, the robot detects a group of two stationary standing pedestrians P1 and P2. The planning results of planner A and planner B are shown in Fig. 6d, where two red dots represent the group of pedestrians. The experimental results show that planner B pursues the minimum path and rudely passes through the group, while planner A detects the group through global pedestrian perception and generates a dynamic costmap so that the planned path meets requirement of society, which verifies the necessity of group interaction modeling.

4.2 Real experiments

- Scene 1: moving pedestrian in the way of robot

  The setting of this scene is similar to scene 1 of simulation, and pedestrians moving in the positive x direc-

Fig. 6 Simulation scenario experiment comparison



(a) planning results of planner A

(b) planning results of planner B

(c) planning failure of planner B

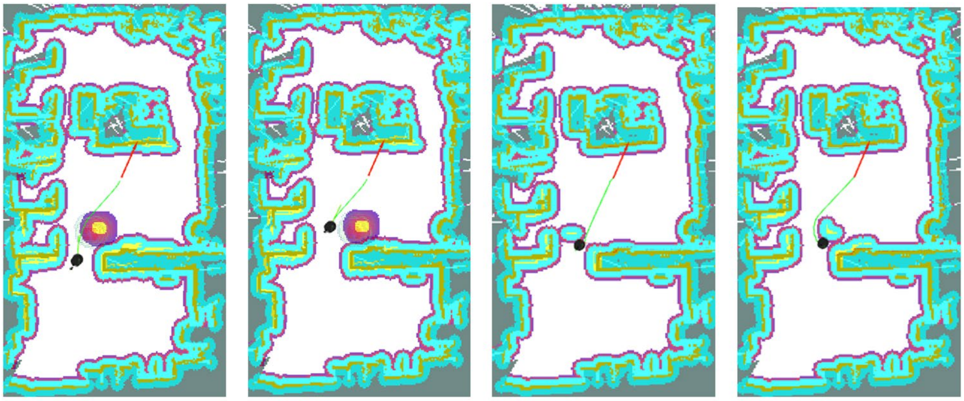
(d) comparison results of scene 2

Table 2 Planner performance

	Planner A	Planner B
Execution time/s	12.1	17.8
Path length/m	4.32	4.60
Shortest distance/m	1.06	0.39

tion are detected during navigation. Planner A generates a dynamic costmap containing pedestrian trajectory information through pedestrian perception and turns to avoid pedestrians in advance, as shown in Fig. 7a and b. Planner B firstly plans the shortest path in the static environment without considering pedestrian because the

Fig. 7 Different time cost map and planning results

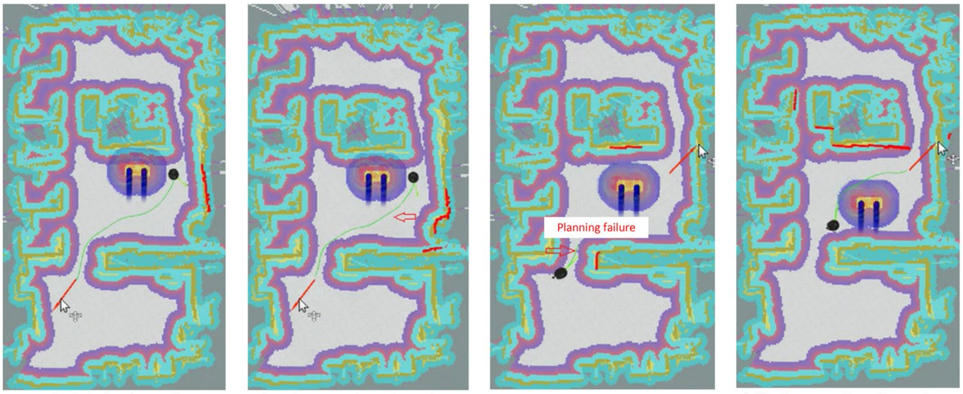


(a) planning results of planner A (1)

(b) planning results of planner A (2)

(c) planning results of planner B (1)

(d) planning results of planner B (2)



(a) initial planning results

(b) dynamic planning results (1)

(c) planning failure

(d) dynamic planning results (2)

Fig. 8 Different time cost map and planning results

global path planning does not take into account the moving pedestrian, as shown in Fig. 7c. As the laser senses the pedestrian and generates the local obstacle cost, the local path planning performs obstacle avoidance shown in Fig. 7d. The experimental results are consistent with the simulation scenario experiments.

- Scene 2: group of pedestrians in the way of robot

  In this scene, two pedestrians walking at roughly the same speed are detected during navigation. Firstly, the moving group is detected by group detection module, after which the dynamic costmap is generated. In the first planning cycle, the planned path shows respect for group and considers group's motion intention. Then, the robot enters the 'Execution' period at the end of the 'Plan' period and performs the initial operation of adjusting direction, as shown in Fig. 8a. At the same time, the planner further optimizes the plan through the next planning cycle. As shown in Fig. 8b, the partial path pointed by the red arrow shortens the path length and steering angles compared to the previous planning cycle. When the target position is occluded by the group, it is easy to cause planning failure in the current planning cycle, as shown in Fig. 8c. At this time, the current achievable optimal state indicated by the red arrow is returned. In the following planning cycle, when the target position is reachable, the path to the target is planned, as shown in Fig. 8d, which further validates the ability of the planning cycle to cope with the dynamic environment.

5 Conclusion

In this paper, a global path planning method based on multi-layer cost map is proposed for human-aware navigation. Based on global-wide pedestrian perception with pedestrian detection and tracking, group interaction and personal space are modelled for generating multi-layer dynamic costmap, which provides social constraint information of the prediction stage for the global path planning. Global path planner defines the cost function and the heuristic cost for heuristic search based on dynamic costmaps and dynamically optimizes the planning result through the 'Plan-Prediction-Execution' cycle. The feasibility of the planner and the superiority to the traditional path planner are verified by various scene experiments and comparative experiments.

Funding This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61573100, 61573101.

References

Bai, H., Cai, S., Ye, N., Hsu, D., Lee, W.S.: Intention-aware online POMDP planning for autonomous driving in a crowd. In: 2015 IEEE international conference on robotics and automation (ICRA). Seattle, USA. IEEE, pp. 454–460 (2015)

Chen, Y.F., Everett, M., Liu, M.: How, Socially aware motion planning with deep reinforcement learning. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). Vancouver, Canada, IEEE, pp. 1343–1350

Escobedo, A., Spalanzani, A., Laugier, C.: Using social cues to estimate possible destinations when driving a robotic wheelchair. In: 2014 IEEE/RSJ International conference on intelligent robots and systems. Chicago, USA: IEEE, pp. 3299–3304 (2014)

Ferrer, G., Sanfeliu, A.: Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments. In: 2014 IEEE/RSJ International conference on intelligent robots and systems. Chicago, USA: IEEE, pp. 1730–1735 (2014)

Ferrer, G., Sanfeliu, A.: Bayesian human motion intentionality prediction in urban environments. Pattern Recogn. Lett. **44**, 134–140 (2014)

Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robot. Autom. Mag. **4**(1), 23–33 (1997)

Gaydashenko, A.: A comparative evaluation of machine learning methods for robot navigation through human crowds. In: 2018 17th IEEE international conference on machine learning and applications. Orlando, USA. IEEE, pp. 553–557 (2018)

Hekmati, A., Gupta, K.: On safe robot navigation among humans as dynamic obstacles in unknown indoor environments. In: 2018 IEEE international conference on robotics and biomimetics. Kuala Lumpur, Malaysia. IEEE, pp. 1082–1087 (2018)

Jiang, L., Wang, W., Chen, Y., Jia, Y.: Personalize vison-based human following for mobile robots by learning from human-driven demonstrations. In: 2018 27th IEEE international symposium on robot and human interactive communication (RO-MAN). Nanjing, China, IEEE, pp. 726–731

Khambhaita, H., Alami, R.:. Assessing the social criteria for human-robot collaborative navigation: a comparison of human-aware navigation planners. In: 2017 26th IEEE international symposium on robot and human interactive communication (RO-MAN). Lisbon, Portugal, IEEE, pp. 1140–1145 (2017)

Kirby, R.: Social robot navigation. Carnegie Mellon University, Pittsburgh (2010)

Kretzschmar, H., Spies, M., Sprunk, C., Burgar, W.: Socially compliant mobile robot navigation via inverse reinforcement learning. Int. J. Robot. Res. **35**(11), 1289–1307 (2016)

Kruse, T., Pandey, A.K., Alami, R., et al.: Human-aware robot navigation: a survey. Robot. Auton. Syst. **61**(12), 1726–1743 (2013)

Lavalle, S., Kuffner, M.: Randomized kinodynamic planning. Int. J. Robot. Res. **20**(5), 378–400 (2011)

Linder, T., Arras, K.O.: Multi-model hypothesis tracking of groups of people in RGB-D data. In: 17th international conference on information fusion (FUSION). Salamanca, Spain: IEEE, pp. 1–7 (2014)

Lu, D. V., Hershberger, D., Smart, W. D.: Layered costmaps for context-sensitive navigation. In: 2014 IEEE/RSJ international conference on intelligent robots and systems. Chicago, USA: IEEE, pp. 709–715 (2014)

Mateus, A., Ribeiro, D., Miraldo, P., et al.: Efficient and robust pedestrian detection using deep learning for human-aware navigation. Robot. Auton. Syst. **113**, 23–37 (2019)

Moussaïd, M., Perozo, N., Garnier, S., et al.: The walking behaviour of pedestrian social groups and its impact on crowd dynamics. PLoS One **5**(4), e10047 (2010)

Park, C., Ondřej, J., Gilbert, M., et al.: HI robot: human intention-aware robot planning for safe and efficient navigation in crowds. In: 2016 IEEE/RSJ international conference on intelligent robots and systems. Daejeon, South Korea: IEEE, pp. 3320–326 (2016)

Pierson, A., Schwarting, W., Karaman, S., Rus, D.: Navigating congested environments with risk level sets. In: 2018 IEEE International conference on robotics and automation (ICRA). Brisbane, Australia: IEEE, pp. 1–8 (2018)

Rios-Martinez, J., Spalanzani, A., Laugier, C.: From proxemics theory to socially-aware navigation: a survey. Int. J. Soc. Robot. **7**(2), 137–153 (2015)

Rösmann C, Feiten W.: Trajectory modification considering dynamic constraints of autonomous robots. In: German Conference on Robotics (ROBOTIK), Munich, Germany, VDE (2012)

Truong, X.T., Ngo, T.D.: Dynamic social zone based mobile robot navigation for human comfortable safety in social environments. Int. J. Soc. Robot. **8**(5), 663–684 (2016)

Truong, X.T., Ngo, T.D.: "To approach humans?": a unified framework for approaching pose prediction and socially aware robot navigation. IEEE Trans. Cognit. Dev. Syst. **10**(3), 557–572 (2017)

Yang, C.-T., Zhang, T., Chen, L.-P., Fu, L.-C.: Socially-aware navigation of omnidirectional mobile robot with extended social force model in multi-human environment. In: 2019 IEEE International conference on systems, man and cybernetics (SMC). Bari, Italy, IEEE, pp. 1962–1968 (2019)

Fang Fang received her Ph.D. degree in School of Automation from the Southeast University, Nanjing, China. She is currently an associate professor in the School of Automation from Southeast University. She has worked in the areas of control and decision of service robot system, robot intelligent perception and social navigation, etc.

Manxiang Shi is studying in School of Automation from the Southeast University, Nanjing, China. He has worked in the areas of robot social navigation and task planning.

Kun Qian received his Ph.D. degree in School of Automation from the Southeast University, Nanjing, China. He is currently an associate professor in the School of Automation from Southeast University. He has worked in the areas of robot intelligent perception and interaction.

Bo Zhou received his Ph.D. degree in Shenyang Institute of automation, Chinese Academy of Sciences. He is currently an associate professor in the School of Automation from Southeast University. He has worked in the areas of Mobile/service robot sensing, navigation and control technology, industrial robot and advanced manufacturing technology, pattern recognition and machine learning, etc.

Yahui Gan received his Ph.D. degree in School of Automation from the Southeast University, Nanjing, China. He is currently a lecturer in the School of Automation from Southeast University. He has worked in the areas of intelligent robot, industrial robot control and advanced manufacturing technology, etc.