Artificial Life and Robotics (2020) 25:264–272
https://doi.org/10.1007/s10015-020-00591-w

ORIGINAL ARTICLE

Safe mobile robot navigation in human-centered environments using a heat map-based path planner

Abhijeet Ravankar[1] · Ankit A. Ravankar[2] · Yohei Hoshino[1] · Michiko Watanabe[1] · Yukinori Kobayashi[2]

Received: 15 April 2019 / Accepted: 7 February 2020 / Published online: 4 March 2020

Abstract
Safe robot navigation in human-centered environments is important to avoid collisions. A major limitation of the traditional path planning algorithms is that the global path is planned only with the knowledge of static obstacles in the map. This paper presents a novel 'HMRP (heat map-based robot path planner)' which uses fixed external cameras to generate a heat map of different passages based on congestion, so that robots can generate congestion-free paths at the global planning stage itself. The congestion values are maintained in a database and the paths are classified into hot and cold regions. Robot navigation is affected by the direction of movement of people. Hence, in this work, the HMRP-based planner also considers the direction of movement of people in passages which improves robot navigation. The proposed HMRP is compared with traditional path planning algorithms in real environment. Results show that the proposed HMRP algorithm generates congestion-free paths for safe robot navigation.

Keywords Robot navigation · Heat map · Path planning

1 Introduction

Autonomous mobile robots are used at many human-centered environments like hospitals, warehouses, and other public places to provide different services like cleaning, surveillance, and delivery. Although many robust path planning algorithms have been proposed, there is always a risk of collision with people in these environments causing injury. Thus, measures to increase safety in mobile robot navigation becomes indispensable to ensure the safety of people.

Path planning includes global and local planning. Once a global path has been planned, the robot starts navigation on it and later avoids collision with people using any of the state-of-the-art algorithms like [12]. Regarding local collision avoidance, the most recent works shows promising results for safe autonomous navigation in indoor [12, 15, 19], and outdoor [10] cases. A review of safe robot navigation is given in [21]. In traditional path planning algorithms, generation of the global path is based on the information of the static obstacles and free space in the map. Once the robot starts navigating on the global path, the local dynamic obstacles are avoided through local planning algorithms which includes trajectory alteration [15, 23] and other approaches like dynamic window approach [1, 7]. The global path generation is mainly based on the shortest path criteria. However, it is possible that the shortest path might be congested. The information about congestion in different paths is unavailable to the robots. A robot could plan a congestion-free path if the congestion information was available to it. The congestion-free path could be slightly longer, however, it would be safer for navigation. In the absence of congestion information, a robot cannot plan trajectories through the congestion-free passages. Hence, this is a limitation. In previous works [20, 22] which also discusses the problems due to unavailability of remote information in the environment, a knowledge-sharing algorithm through which the robots can share information about the new obstacles and congestion in the map with other robots has been proposed. However, the global path generation was still based on the shortest path

This work was presented in part at the 24th International Symposium on Artificial Life and Robotics, Beppu, Oita, January 23–25, 2019.

✉ Abhijeet Ravankar
aravankar@mail.kitami-it.ac.jp

[1] Faculty of Engineering, Kitami Institute of Technology, Hokkaido, Japan

[2] Division of Human Mechanical System and Design, Hokkaido University, Sapporo, Japan

criteria and information sharing was only possible when a robot actually visits particular areas of the map which is not guaranteed. Hence, lack of congestion information on different paths adversely affects path planning.

In previous works, many researchers have addressed the problem of robot navigation in human-centric environments using external sensors. Nitta et al. [4] proposed an algorithm for robot path planning which uses pedestrian information map which is generated from human trajectories logged using sample-based join probabilistic data association filters with a laser range finder (LRF). Similarly, anticipatory robot path planning in human environments is proposed in [4] which tries to give anticipation ability to a robot by simulating people's reaction to robot's motion during planning. They also use a torso-level Hokuyo LRF scanner for person tracking. Laser range finder is used for the detection and tracking of human legs for person identification in environment in [3], while an environmental map describing the walking activity of people including walking dynamics is proposed in [13] using LRF sensor. Visual sensors have been in used in [24], in which the human walking trajectories are obtained from distributed sensors and frequently used paths in the environment are extracted. Similarly, in [25], human path patterns are extracted for mobile robot navigation using distributed sensors in which the global position of the mobile robot can be directly measured using external sensors, which makes the localization problem. The localization problem is also addressed in [16, 31] through optimal placement of passive sensors in the environment. These approaches have a merit of reducing the sensors on the robot side, thereby reducing the cost, weight and probability of hardware faults. On the other hand, some researchers have addressed the problem of mobile robot navigation in human-centric environments by modeling social force model (SFM) [30] in robot's navigation. In [26], authors have dealt with situation where two walkers have crossing trajectories. Based on these human trajectories involving a collision avoidance task, the total effort is determined and shared between each walker depending on several factors of the interaction such as crossing angle, time to collision and speed. In [2], authors have proposed a planning framework that uses communication during mobile robot navigation in a manner that humans find understandable using wearable haptic interface and the vibration motor. Social navigation model based on human intention analysis using face orientation is proposed in [17], in which, the trajectories of humans are classified based on the face orientation on a social force model and their predicted motion.

However, most of the previous approaches use LRF sensors which are expensive and setting multiple LRFs in the environment is not always feasible. Comparatively, visual sensors like cameras are not expensive to setup and many buildings already have security cameras installed. Previous works employing visual sensors usually work with grid maps to estimate people's trajectories. However, directly working with individual grid coordinates of the map results in enormous space complexity and communicating the raw trajectory information to robots is expensive in terms of bandwidth. To overcome these limitations, this paper presents a novel HMRP: a heat map-based robot path planner for service robots. A heat map is generated from the cameras fixed in different passages and areas that track the net influx of people, in different times. Based on this historical data maintained in a database, the paths are classified into various categories based on the congestion at different times. The passages with large movement of people are the hot portions of the map. On the other hand, passages with less movement of people are the cold areas. The hot and cold areas of the map vary with time. Access to these hot and cold areas enables a robot to plan better paths through less congested areas. The local collision avoidance is done by traditional algorithms [12]. However, with the proposed HMRP algorithm, the dynamic obstacle avoidance algorithms also benefit as the selected path has less number of dynamic obstacles. This also ensures a faster service time and safe navigation. The space complexity problem is addressed using a novel 'node map' representation which drastically reduces the amount of data to be communicated to the robots.

Traditionally, information from external cameras has been used mainly for surveillance and monitoring. The proposed work uses fixed external cameras to detect congestion in passages and incorporate this information in robot's path planning. The gist of novel contribution of the proposed work lies in camera–robot network for congestion-free path planning. The novel contributions of the proposed work are summarized below:

1. The proposed work uses fixed cameras in corridors to estimate congestion. The proposed work also estimates the heap value based on the direction of movement of people in corridors.
2. Fixed camera-based heat map estimation in the proposed work is done while considering the direction of traffic flow. This has a significant improvement on path planning from a practical point of view.
3. A way to incorporate the heat map's value in grid-based path planning is proposed by altering movement cost.
4. A node map is proposed which simplifies communication and fusing congestion information in robot's path planning.
5. The proposed work uses historical congestion information in robot path planning for situations when real-time congestion information is not available.

Thus, robots are able to plan better paths through less congested passages which results in safer navigation.

2 Generation of grid map

To quantitatively estimate the congestion in different parts of the service area, it is important to first construct a grid map and thereby a node map of the environment. A grid map discretizes the world into cells. Each cell is a binary random variable that models if a cell m_i is occupied $p(m_i) \rightarrow 1$, free $p(m_i) \rightarrow 0$, or unknown $p(m_i) \rightarrow 0.5$. It is assumed that the map is static and the cells are independent. The probability distribution of the map is given by the product over the cells, $p(m) = \prod_i p(m_i)$. Given sensor data $z_{1:t}$ and the poses $x_{1:t}$ of the sensor, the goal is to estimate the map,

$$p(m|z_{1:t}, x_{1:t}) = \prod_i p(m_i|z_{1:t}, x_{1:t}). \tag{1}$$

```
Algorithm 1: Grid-mapping
1  Function grid_mapping($\{l_{t-1,i}\}, x_t, z_t$)
2      if m_i in range z_t then
3          $l_{t,i} \leftarrow l_{t-1,i} + \mathrm{ism}(m_i, x_t, z_t) - l_0$
4      else
5          $l_{t,i} \leftarrow l_{t-1,i}$
6      return($l_{t,i}$)
   ...
7  Function ism($m_i, x_t, z_t$)
8      $r \leftarrow \sqrt{(x_i - x)^2 + (y_i - y)^2}$
9      $\phi \leftarrow \tan^{-1}\left(\frac{y_i - y}{x_i - x}\right) - \theta$
10     $k \leftarrow \mathrm{argmin}_j |\phi - \theta_{j,sens}|$
11     if $r > \min(z_{\max}, z_t^k + \frac{\eta}{2})$ or $|\phi - \theta_{k,\mathrm{sens}}| > \frac{\Gamma}{2}$ then
12         return($l_0$)
13     if $z_t^k < z_{\max}$ and $|r - z_t^k| < \frac{\eta}{2}$ then
14         return($l_{\mathrm{occupied}}$)
15     if $r \leq z_t^k$ then
16         return($l_{\mathrm{free}}$)
```

The pseudo-code for grid map generation is shown in Algorithm 1. It uses an inverse measurement model (Algo.1, Line 7) for robots equipped with range finders. The robot pose is supposed to be $(x, y, \theta)^{\mathrm{T}}$. Lines 7–16 calculate the inverse model [29] by first determining the beam index k and the range r for the center-of-mass of the cell m_i. Moreover, for ease of computation, log-odds ratio $l_{t,i}$ and l_0 are used and defined as,

$$\begin{aligned} l_{t,i} &= \log\frac{p(m|z_{1:t}, x_{1:t})}{1 - p(m|z_{1:t}, x_{1:t})}, \\ l_0 &= \log\frac{p(m_i = 0)}{p(m_i = 1)} = \log\frac{p(m_i)}{1 - p(m_i)}. \end{aligned} \tag{2}$$

The algorithm for grid map generation is given in Algorithm 1, in which, η is the thickness of obstacles, and Γ is the width of a sensor beam. These are sensor-specific parameters and other algorithms for grid mapping can also be used [29].

3 Generation of node map

```
Algorithm 2: Node-map Generation
   Data: m : Gridmap, m_height : map height, m_width : map width
1  Function node_mapping(m)
2      for each row in m_height do
3          for each col in m_width do
4              if cell m[row][col] is unknown then
5                  $m[row][col] \leftarrow l_{\mathrm{occupied}}$
6  Successively erode and dilate binary image [8]
7  Apply skeletonization algorithm
8  Apply prunning algorithm
9  Detect lines segments and their endpoints [18]
10 Cluster nearby endpoints in range $\delta$ with k means
11 Mark clustered points as nodes N ← $\{n_1, n_2, \cdots, n_m\}$
12 return(N)
```

We define '`node`' as a point of turn in a path of the map. All the paths of the map are represented as a network of these nodes. Figure 1 shows the node representation of the path. The nodes $n_1, n_2, \ldots, n_7$ are the points of turns in the map. The pseudo-code for generating a node map is given in Algorithm 2. The input to the algorithm is a grid map with obstacles (black), open (white), and unknown (grey) areas. The first step is to generate a binary image of the grid map which is done by turning all unknown cells m_i to blocked (black) value, indicated by lines 2–5 of Algorithm 2. Noise is removed by successively applying morphological erode and dilate operations [8]. The next step is to apply skeletonization algorithm [32]. For example, skeletonization of binary map in Fig. 2a is shown in Fig. 2b. Many skeletonization and thinning algorithms generate unnecessary tentacles which needs to be removed using pruning algorithm. Line segments are then detected using techniques like SVD or Hough Transform [18] (Fig. 2c). The end points of segments which are within a small threshold distance δ as shown in Fig. 2d are clustered into a single node 'n' as shown in Fig. 2e. A graph 'm' of these nodes $\{n_1, n_2, \ldots, n_m\}$ form the node map of the environment.

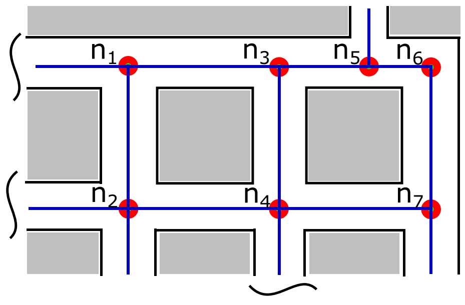


Fig. 1 Node representation of map

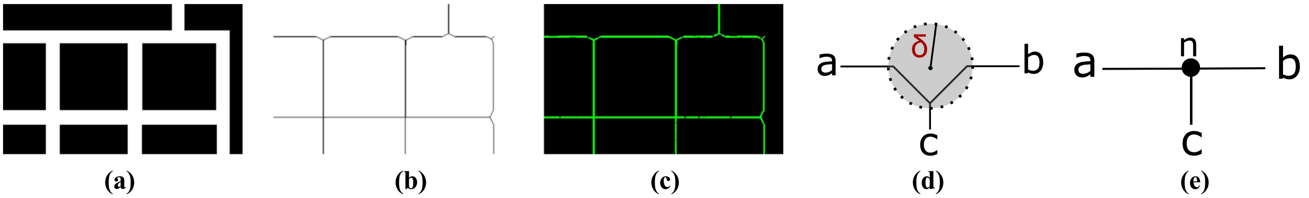


Fig. 2 Node map generation. **a** Binary map. **b** Skeleton map. **c** Hough lines. **d** Cluster within δ. **e** Clustered node n

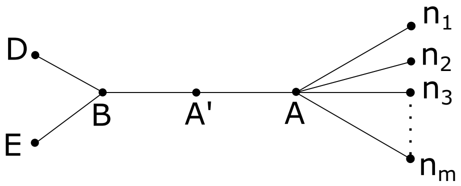


Fig. 3 Influx $\Phi(\cdot)$ at nodes and sub-nodes

The advantages of using a node map instead of a grid map are:

1. The task of placing external cameras on the nodes becomes easy. Moreover, depending on the field-of-view (FOV) of the camera, 'sub-nodes' can be inserted between two nodes which are placed at a large distance.
2. It becomes easier to discretize the flow between various portions of the map. Moreover, these discrete segments can later be correlated to various time zones, which will be explained in Sect. 4.
3. Mathematically modeling the net influx of people from other nodes becomes easier.
4. Congestion across the edges of the node map require less memory for storage and can quickly be communicated to the robots as against previous works which work on individual grid pixel-based heat map estimation.

Figure 3 shows several nodes with a sub-node. If $\Phi(\cdot)$ represents the net influx, then two corollaries can be given.

Corollary 1 *If A' is a sub-node between nodes A and B, then $\Phi(\overrightarrow{BA'}) = \Phi(\overrightarrow{A'A})$, and $\Phi(\overrightarrow{AA'}) = \Phi(\overrightarrow{A'B})$ (Fig. 3)*

Proof of Corollary 1 is straightforward from the property of sub-node which is defined as an intermediate point with no turn. Hence, from Fig. 3, $\Phi(A') = 0$.

Corollary 2 *The net directional influx at node is equal to the sum of directional influxes from other nodes joining that node. From Fig.3,*

$$\Phi(\overrightarrow{AB}) = \sum_{1}^{m} \Phi(\overrightarrow{n_iA}) - \sum_{1}^{m} \Phi(\overrightarrow{An_i}). \tag{3}$$

Proof of Corollary 2 follows from Corollary 1 and the assumption that directional flow of entity does not change between two nodes, and $\Phi(\overrightarrow{AB}) \neq \Phi(\overrightarrow{BA})$.

4 Congestion detection with fixed external camera

The fixed cameras are placed at several nodes in the map to detect people in passages. Once a person's position in the camera image is found, we can calculate its position in the world coordinate system through camera calibration. It defines relation between image and world coordinate system using a pinhole camera model shown in Fig. 4. This relationship is given by,

$$m = A\ [\ R\ |\ t\ ]\ M', \tag{4}$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{5}$$

where A is an upper triangular intrinsic parameter matrix with (f_x, f_y) as the focal lengths in pixels, and $[c_x, c_y]^T$ is the principal point at the center of the image. Extrinsic parameter matrix $[R|t]$ comprises rotation matrix (R) and translation vector (t). HMRP assumes that the camera is fixed. $[x_w, y_w, z_w]^T$ are the coordinates of the robot in world space which are mapped to the image coordinates $[u, v]^T$ in pixels.

Although there are different ways to detect moving people in images, this work uses a method of foreground pixel detection from a robust background. A robust background $\mathbf{BG} \in \mathbb{R}^2$ is first generated from current camera frame $\mathbf{I}_t \in \mathbb{R}^2$ at time 't' to cope with illumination changes and obstruction by static obstacles in the corridor. For the purpose of robustness, two or more background images are generated using different learning rates λ_i, as given in Eq. 6.

Table 1 Database of heat value $\mathbb{H}_i(\mathbf{E}_i)$ considering direction. Congested paths are marked in bold and free paths in italic

Date (day)	Time interval	$\mathrm{E}(\overrightarrow{n_1 n_2})$	$\mathrm{E}(\overrightarrow{n_2 n_3})$	$\mathrm{E}(\overrightarrow{n_3 n_4})$	⋯	$\mathrm{E}(\overrightarrow{n_4 n_1})$
23-10-2018 (Tues)	11:00-11:30	59	*3*	*18*	⋯	*14*
	11:30-12:00	**103**	*10*	37		33
	12:00-12:30	**112**	35	52		**88**
	12:30-13:00	**95**	37	67		**84**
	… …	…	…	…	…	…
24-10-2018 (Wed)	12:00-12:30	**125**	*23*	43		65
	12:30-13:00	**102**	*12*	69		71
	… …	…	…	…	…	…
25-10-2018	… …	…	…	…	…	…

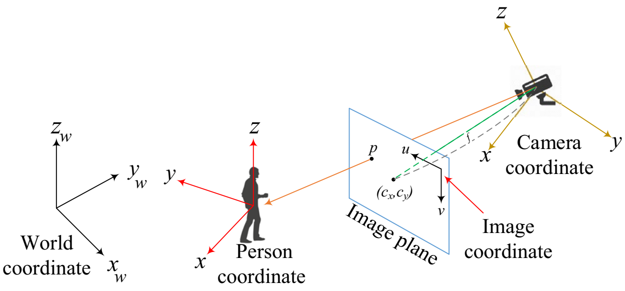


Fig. 4 Camera and world frame

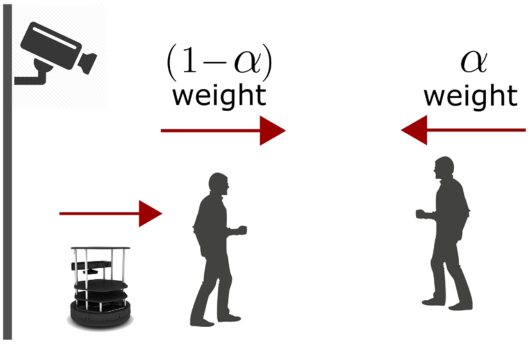


Fig. 5 Influence of direction on weight. Collision avoidance of people coming from opposite direction is more difficult

$$\begin{aligned}\mathbf{BG}_{\text{slow}}(x,y) &= \lambda_{\text{slow}}\ \mathbf{I}_t(x,y) \oplus (1-\lambda_{\text{slow}})\ \mathbf{I}_{t-1}(x,y),\\ \mathbf{BG}_{\text{fast}}(x,y) &= \lambda_{\text{fast}}\ \mathbf{I}_t(x,y) \oplus (1-\lambda_{\text{fast}})\ \mathbf{I}_{t-1}(x,y).\end{aligned} \tag{6}$$

The coordinates of the moving entities $\mathbf{T}_{\text{p}}(x,y) \in \mathbb{R}^2$ are computed by probabilistic difference between the current frame $\mathbf{I}_t$ and the two background images $\mathbf{BG}_{\text{slow}}$, and $\mathbf{BG}_{\text{fast}}$. All the pixels where the difference value is more than Υ are marked as moving foreground pixels, as given in Eq. 7.

$$\mathbf{T}_{\text{p}} \leftarrow f(\mathbf{I}_t, \mathbf{BG}_{\text{slow}}, \mathbf{BG}_{\text{fast}}, \Upsilon). \tag{7}$$

4.1 Mapping grid pixel to vertex with directional influence and profile generation

We define a function $\xi(\cdot)$ (Eq. 8) which maps the coordinates of $\mathbf{T}_{\text{p}}(x,y)$ to the nearest edge of the node map. This mapping is achieved through a normal $\perp$ from the edge $(\mathbf{E}_{\text{i}})$ to the detected hot-pixel coordinate. The heat value of the edge $(\mathbf{E}_{\text{i}})$ given by $\mathbb{H}(\mathbf{E}_{\text{i}})$ is increased proportional to the total hot pixels mapped. For each detection of a moving person, multiple hot pixels may be mapped to the edge. The direction of movement of people influences robot navigation. As shown in Fig. 5, robot navigation is relatively easy if it is moving in the same direction as it can safely follow the person. However, robot trajectory needs to be altered to avoid collision with person approaching from the opposite direction. This effect is controlled by weight factor α $(0 < \alpha < 1)$ which is set to a large value for person approaching from the opposite direction. The direction is estimated by optical flow or increasing pixels.

$$\xi(\mathbf{T}_{\text{p}} \mapsto \mathbf{E}_i,\ \mathbb{H}_i, \kappa, \alpha), \qquad \xi : \mathbb{R}^2 \to \mathbb{R}, \tag{8}$$

where κ is a constant weight used to control the effect of increase in edge hotness from 'n' hot pixels. The $\mathbb{H}$ value is calculated by Eq. 9 using the `ceil` function,

$$\mathbb{H}_i^{t+1} \leftarrow \mathbb{H}_i^t + \left\lceil \alpha \cdot \frac{n}{\kappa} \right\rceil. \qquad \mathbb{H}_i \to \mathbb{R} \tag{9}$$

The parameter α incorporates the direction of traffic and affects $\mathbb{H}_i$ which later affects the obstacle weights in the grid map. Hence, path planning is also improved comparatively.

A profile for each edge $\mathbf{E}_i$ is generated in intervals of period $\varDelta T$ by accumulating the heat value $(\mathbb{H}_i)$ of each edge in the node map (in Table 1, $\varDelta T$ = 30min). This data are stored in a dictionary $\mathbb{D}$ (Eq. 10) which maintains the historical data of the heat values across each edge. The dictionary stores data in {`key:value`} pair. A robot can access this dictionary through a key $\mathbf{E}_i$ to get the corresponding heat value ($\mathbb{H}_i$) which gives an estimation about the crowd in that passage.

$$\mathbb{D} \equiv \left\{ \mathbf{E}_i, \ : \ \sum_{t=1}^{\Delta T} \mathbb{H}_i^{t+T} \left(\mathbf{E}_i \right) \right\}. \tag{10}$$

Table 1 shows the database of heat value $\mathbb{H}_i(\mathbf{E}_i)$ at various nodes in a time interval (ΔT) of 30 min. Congested paths with heat value $\mathbb{H} \geq 80$ are shown in red. These edges mark congested portions of the map during the respective time.

5 Path planning on heat map

In this section, we discuss how congestion information is fused in path planning. Our implementation uses A* algorithm whose details can be found in [5, 9, 11]. Any other path planning algorithm which uses a costmap for movement in a map like D* algorithm [28], Focussed D* [27], or Dijkstra's algorithm [6] can also be used. The heat value $\mathbb{H}_i$ is reflected in path planning by increasing the robot's movement cost in corresponding grids. The cost W_{i}^b of movement of robot in the map is set beforehand. In case of a grid map, the cost is generally set to 1 unit in perpendicular directions, and $\sqrt{2}$ units in diagonal directions, in free space. Congestion information is fused by increasing the cost of movement in grids with high congestion. The proposed work uses a node map with edges $\mathbf{E}_i$ for efficiency. HMRP increments the cost by a weighted average $\widehat{W}_{\text{i}}^c$ of the corresponding current heat value $\mathbb{H}_i^t$ and the average of previous value $\overline{\mathbb{H}_i^{t-1}}$ (Eq. 11). If the current value is not available, ρ is set to 0. We briefly explain the A* algorithm-based path planning. Let G = (V,E) is a graph with non-negative edge distances, and h(·) is an admissible heuristic. Let S be the start location and G be the goal node of the robot. If d(v) is the shortest distance from S to v seen so far, then d(v) + h(v) gives an estimate of the distance from S to v and similarly from v to G. The queue of nodes $\text{Q}_\text{h} = (\text{V}_1, \text{V}_2, \cdots, \text{V}_\text{n})$ sorted by d(v) + h(v) is the A* path from S to G. The incremented cost $W_{\text{i,obs}}^{+}$ is used to calculate the shortest path. The cost of movement across grids along an edge of higher congestion is increased. Thus, costmap-based planner estimates paths with lower congestion.

$$\begin{aligned} \widehat{W}_{\text{i}}^c &\leftarrow \rho \ \mathbb{H}_i^t(\mathbf{E}_i) + (1-\rho)\overline{\mathbb{H}_i^{t-1}(\mathbf{E}_i)}, \\ W_{\text{i,obs}}^{+} &\leftarrow W_{\text{i}}^b + \widehat{W}_{\text{i}}^c \end{aligned} \tag{11}$$

6 Experiments and results

Experiments were carried in real environment to test HMRP. Pioneer-P3DX (Fig. 6a) was used, which was equipped with distance sensors (Microsoft Kinect and UHG-08LX laser range sensor) and cameras. The experiment environment is shown in Fig. 7a. Raspberry-Pi 3 model B+ was used for external camera node which features a 1.4GHz 64-bit quad-core ARM Cortex-A53 CPU with 1GB RAM. The camera node was operated at 10 FPS with VGA resolution. The camera was set at a height of 238 cm above the ground. Cameras should be setup at considerable height so that a complete occlusion of camera by obstacles is improbable. The distance sensor is accurate within ±30 mm within 1 m and within 3% of the detected distance between 1 and 8 m. The angular resolution is approximately 0.36 degrees. We first describe the motion model of the robot. The distance between the left and the right wheel is W_r and the robot state at position P is given as $[x, y, \theta]$. From Fig. 6b, turning angle $\beta \neq 0$ and radius of turn R is calculated as,

$$r = \beta \cdot (R + W_r), \ l = \beta \cdot R, \ \therefore \beta = \frac{r - l}{W_r}, \ R = \frac{l}{\beta}. \tag{12}$$

The coordinates of the center of rotation (C, in Fig. 6b), and new heading θ' are calculated as,

$$\begin{bmatrix} Cx \\ Cy \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \left(R + \frac{W_r}{2} \right) \cdot \begin{bmatrix} \sin\theta \\ -\cos\theta \end{bmatrix}, \ \theta' = (\theta + \beta) \text{mod} 2\pi, \tag{13}$$

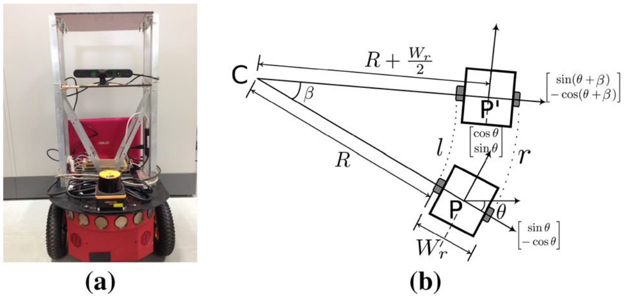
from which the coordinates of the new position P' are calculated as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} Cx \\ Cy \end{bmatrix} - \left(R + \frac{W_r}{2} \right) \cdot \begin{bmatrix} \sin\theta' \\ -\cos\theta' \end{bmatrix}, \beta \neq 0 \implies r \neq l. \tag{14}$$

If $r = l$, i.e., if the robot motion is straight, the state parameters are given as $\theta' = \theta$, and,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + l \cdot \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, (l = r). \tag{15}$$

The experiments were set without real-time camera $\mathbb{H}_i$ information and ρ in Eq. 11 was set to 0 using only previous data. α in Eq. 9 was set to 0.6. The grid map with nodes


C
$R + \frac{W_r}{2}$
β
R
P'
l
r
$\begin{bmatrix} \sin(\theta+\beta) \\ -\cos(\theta+\beta) \end{bmatrix}$
$\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$
P
θ
$\begin{bmatrix} \sin\theta \\ -\cos\theta \end{bmatrix}$
W_r

(a) (b)

Fig. 6 Real experiment setup. **a** Pioneer P3DX, **b** motion model

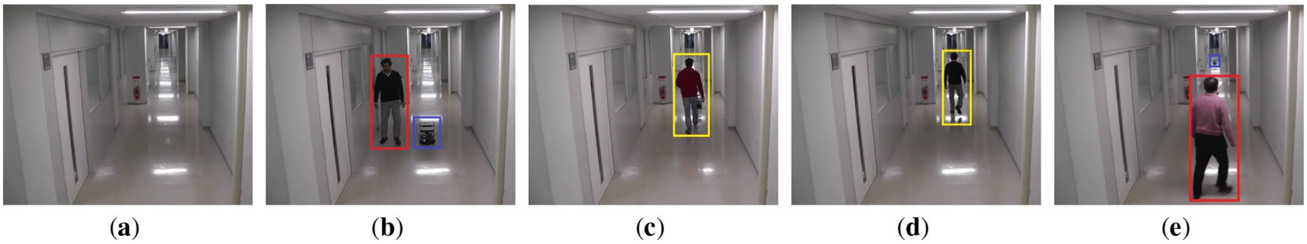


Fig. 7 Congestion detection using external camera. **a** Passage environment. (b ∼ e) People detection

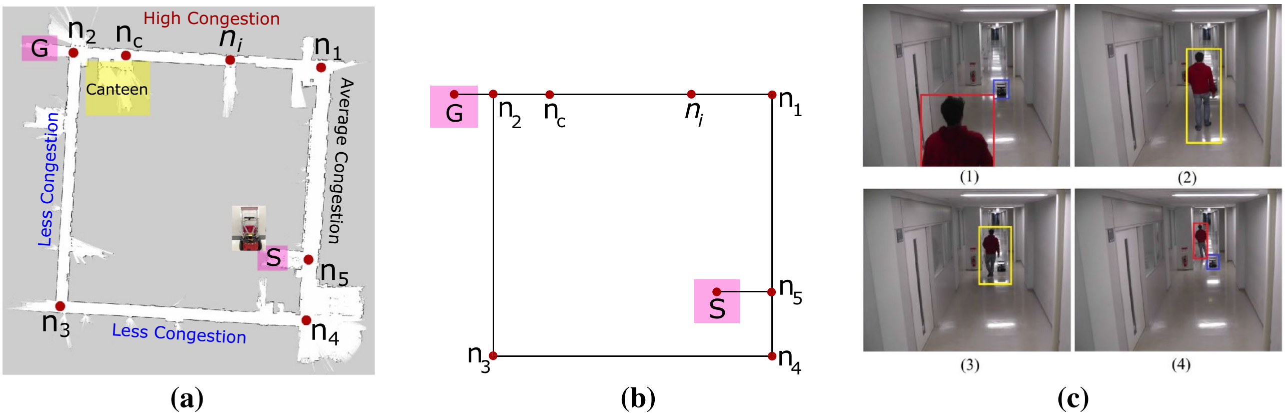


Fig. 8 Real experiment setup. **a** Pioneer P3DX, **b** motion model. **c** Grid map of the environment of Fig. 7a. Robot's start (S) and goal (G) locations, and cafeteria are marked. **d** Node map. **e** People moving in the direction of robot

$n_1, \cdots, n_5$ is shown in Fig. 8a. The resolution of grid map was set to 0.1 m per occupancy grid. The equivalent node map is shown in Fig. 8b in which n_i is an intermediate node. Robot's start (S), and goal (G) locations are marked along with the cafeteria. The experiment was conducted during the peak lunch time of university (12:30–13:00) to test path planning in congestion. The alternate paths of the environment had relatively less congestion during this time. A* algorithm [9] was chosen for path planning, however, any other algorithm can also be chosen. The grid-based navigation had one unit cost for forward, back, left, and right movement, whereas, for diagonal movement the cost was $\sqrt{2}$ units. Table 2 summarizes the paths planned with traditional and proposed method. Table 3 shows the results of $\mathbb{H}$-score in different passages of Fig. 8a in which the highly congestion and relatively free paths are marked. Figure 8c shows a scenario in which the person is moving in the same direction as that of robot. The shortest path length without HMRP was 2042 units, whereas the path length with HMRP was 2491 units. Figure 7b–e shows some results of people detection in passages. Notice that HMRP path is 21.9% longer than the shortest path. However, the shorter path had more congestion, while the HMRP path had less congestion and therefore safer for robot navigation. Since the same dimensions of the costmap was used for path planning using the A* algorithm, the computation time for both traditional and the proposed algorithm was the same. The overhead of fusing congestion information in the costmap in the proposed algorithm was 1.3 s for the gridmap shown in Fig. 8a. Since global path is planned once during the start of navigation of the robot, this overhead time is minuscule.

Table 2 Generated paths with and without virtual obstacles

Method	Path	Path length
Traditional	S → n_5 → n_1 → n_2 → G	2042
Proposed	S → n_5 → n_4 → n_3 → n_2	2491

Table 3 Effect of movement direction on $\mathbb{H}$-score

Method	Path	$\mathbb{H}$-score
HMRP	$\overrightarrow{n_4n_1} + \overrightarrow{n_1n_2}$	174
	$\overrightarrow{n_5n_4} + \overrightarrow{n_4n_3} + \overrightarrow{n_3n_2}$	96

The proposed node map simplifies communicating information with the robot. If the dimensions of the map are $W_m \times H_m$, where W_m and H_m are the width and height, respectively, and the map is divided into small grids of dimensions $\delta_w \times \delta_h$, the total number of small grids (N_{grids}) in the map are,

$$N_{\text{grids}} = \frac{W_m \times H_m}{\delta_w \times \delta_h}. \tag{16}$$

Hence, in previous works [4, 14], a huge amount of congestion information proportional to $N_{\text{grids}} = \frac{W_m \times H_m}{\delta_w \times \delta_h}$ at the level of individual grid pixels needs to be communicated to the robots, which is not efficient. The proposed approach uses a node map with edges over which the heat values represents congestion and hence there is a scalar value over each edge. In the proposed work, the amount of information communicated to the robots is proportional to N_{edges}, where N_{edges} is the total number of edges in the map which are few and $N_{\text{edges}} << N_{\text{grids}}$. Hence, information is communicated quickly in the form of a dictionary ($\mathbb{D}$) of key-value pair where the keys are the edges and the values represent the heat value or congestion as $\mathbb{D} = \{(\mathbb{E}_1 : \mathbb{H}_1), (\mathbb{E}_2 : \mathbb{H}_2), \cdots, (\mathbb{E}_{N_{\text{edges}}} : \mathbb{H}_{N_{\text{edges}}})\}$. Due to the same simple representation in a dictionary form, there is no need for a graphical representation showing hot and cold regions with different colors over the map as the values are summarized succinctly. Moreover, fusing the congestion information in path planning is straightforward in the proposed algorithm which uses node map. Also note that previous works consider only the current congestion information for path planning which is discarded once it is used. However, real-time information might not be always available to the robot as real-world scenarios has problems of sensor failure, communication failure, etc. The proposed work uses both the current and historical information and the current information is prioritized whenever available. The historical information also enables a robot to 'plan ahead' its queued tasks, in future time. Further, previous works like [14] uses pedestrian classification using sample-based joint probabilistic data association filters and Gaussian mixture model which requires labeling and training. While, no data labeling or training is required in the proposed work which uses direct detection using cameras. In addition, previous works [4, 14] uses expensive Lidar sensors in the environment. Installation of such sensors is not feasible or impractical in terms of power supply, maintenance, cost, and scalability. On the other hand, the proposed approach uses inexpensive cameras in the corridors. Most buildings have security cameras installed in passages and image data from such cameras can directly be used to estimate congestion in passages and efficient path planning of robots.

7 Conclusion

In this paper, we proposed a new HMRP planner which takes the influx of people in passages calculated from fixed external cameras into consideration while planning safe robot paths. A grid and node map are formulated to estimate crowd in different sections of the map. Later, moving pixels detected in the passages are mapped to the corresponding edges increasing its heat value. The movement cost on the edges with large crowd is increased. Thereafter, any traditional planner can be used to estimate the shortest path. Robot navigation is affected by the direction of movement of people. Therefore, we improved the idea by also considering the direction of movement of people in increasing the movement cost in grids. With the inclusion of the current and historical crowd influx factor, HMRP estimates the paths which are less crowded and safe for navigation. A major benefit of HMRP is that it can directly use the building or shop's infrastructure which generally has fixed external cameras. In future, we plan to improve the algorithm by considering the width of the passages and test the algorithm in very narrow corridors. Estimation of congestion using on-board cameras fixed on the robot is also considered as future work.

References

1. Brock O, Khatib O (1999) High-speed navigation using the global dynamic window approach. In: Proceedings 1999 ieee international conference on robotics and automation (Cat. No.99CH36288C), p 1, vol 1. https://doi.org/10.1109/ROBOT.1999.770002
2. Che Y, Okamura AM, Sadigh D (2018) Efficient and trustworthy social navigation via explicit and implicit robot-human communication. arXiv:1810.11556
3. Chung W, Kim H, Yoo Y, Moon C, Park J (2012) The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder. IEEE Trans Ind Electron 59(8):3156–3166. https://doi.org/10.1109/TIE.2011.2170389
4. Cosgun A, Sisbot EA, Christensen HI (2016) Anticipatory robot path planning in human environments. In: 2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN), pp 562–569. https://doi.org/10.1109/ROMAN.2016.7745174
5. Delling D, Sanders P, Schultes D, Wagner D (2009) Engineering route planning algorithms. In: Lerner J, Wagner D, Zweig K (eds) Algorithmics of large and complex networks, lecture notes in computer science, vol 5515. Springer, Berlin, Heidelberg, pp 117–139. https://doi.org/10.1007/978-3-642-02094-0_7
6. Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1(1):269–271
7. Fox D, Burgard W, Thrun S (1997) The dynamic window approach to collision avoidance. IEEE Robot Autom Mag 4(1):23–33. https://doi.org/10.1109/100.580977
8. Haralick R, Sternberg SR, Zhuang X (1987) Image analysis using mathematical morphology. IEEE Trans PAMI Pattern Anal Mach Intell 9(4):532–550. https://doi.org/10.1109/TPAMI.1987.4767941

9. Hart P, Nilsson N, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. Syst Sci Cybern IEEE Trans 4(2):100–107. https://doi.org/10.1109/TSSC.1968.300136
10. Koschi M, Pek C, Beikirch M, Althoff M (2018) Set-based prediction of pedestrians in urban environments considering formalized traffic rules. In: 2018 21st international conference on intelligent transportation systems (ITSC), pp 2704–2711. https://doi.org/10.1109/ITSC.2018.8569434
11. LaValle SM (2006) Planning algorithms. Cambridge University Press, Cambridge, UK. http://planning.cs.uiuc.edu/. Accessed 11 Feb 2016
12. Liu SB, Roehm H, Heinzemann C, Lütkebohle I, Oehlerking J, Althoff M (2017) Provably safe motion of mobile robots in human environments. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 1351–1357. https://doi.org/10.1109/IROS.2017.8202313
13. Nishio T, Niitsuma M (2019) Environmental map building to describe walking dynamics for determination of spatial feature of walking activity. In: 2019 IEEE 28th international symposium on industrial electronics (ISIE), pp 2315–2320. https://doi.org/10.1109/ISIE.2019.8781155
14. Nitta J, Sasaki Y, Mizoguchi H (2015) Path planning using pedestrian information map for mobile robots in a human environment. In: 2015 IEEE international conference on systems, man, and cybernetics, pp 216–221. https://doi.org/10.1109/SMC.2015.50
15. Penin B, Giordano PR, Chaumette F (2019) Minimum-time trajectory planning under intermittent measurements. IEEE Robot Autom Lett 4(1):153–160. https://doi.org/10.1109/LRA.2018.2883375
16. Rakhim B, Zhakatayev A, Adiyatov O, Varol HA (2019) Optimal sensor placement of variable impedance actuated robots. In: 2019 IEEE/SICE international symposium on system integration (SII), pp 141–146. https://doi.org/10.1109/SII.2019.8700432
17. Ratsamee P, Mae Y, Ohara K, Kojima M, Arai T (2013) Social navigation model based on human intention analysis using face orientation. In: 2013 IEEE/RSJ international conference on intelligent robots and systems, pp 1682–1687. https://doi.org/10.1109/IROS.2013.6696575
18. Ravankar A, Ravankar AA, Hoshino Y, Emaru T, Kobayashi Y (2016) On a hopping-points svd and hough transform based line detection algorithm for robot localization and mapping. Int J Adv Robot Syst 13(3):98. https://doi.org/10.5772/63540
19. Ravankar A, Ravankar AA, Kobayashi Y, Emaru T (2016) Path smoothing extension for various robot path planners. In: 2016 16th international conference on control, automation and systems (ICCAS), pp 263–268. https://doi.org/10.1109/ICCAS.2016.7832330
20. Ravankar A, Ravankar A, Kobayashi Y, Emaru T (2017) Symbiotic navigation in multi-robot systems with remote obstacle knowledge sharing. Sensors 17(12):1581. https://doi.org/10.3390/s17071581
21. Ravankar A, Ravankar A, Kobayashi Y, Hoshino Y, Peng CC (2018a) Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. Sensors 18(9):3170. https://doi.org/10.3390/s18093170
22. Ravankar A, Ravankar A, Kobayashi Y, Hoshino Y, Peng CC, Watanabe M (2018b) Hitchhiking based symbiotic multi-robot navigation in sensor networks. Robotics 7(3):37. https://doi.org/10.3390/robotics7030037
23. Ravankar A, Ravankar A, Rawankar A, Hoshino Y, Kobayashi Y (2019) Itc: Infused tangential curves for smooth 2d and 3d navigation of mobile robots. Sensors 19(20):4384. https://doi.org/10.3390/s19204384
24. Sasaki T, Hashimoto H (2006) Human observation based mobile robot navigation in intelligent space. In: 2006 IEEE/RSJ international conference on intelligent robots and systems, pp 1044–1049. https://doi.org/10.1109/IROS.2006.281808
25. Sasaki T, Brscic D, Hashimoto H (2010) Human-observation-based extraction of path patterns for mobile robot navigation. IEEE Trans Ind Electron 57(4):1401–1410. https://doi.org/10.1109/TIE.2009.2030825
26. Silva G, Olivier A, Crétual A, Pettré J, Fraichard T (2018) Human inspired effort distribution during collision avoidance in human-robot motion. In: 2018 27th IEEE international symposium on robot and human interactive communication (RO-MAN), pp 1111–1117. https://doi.org/10.1109/ROMAN.2018.8525623
27. Stentz A (1995) The focussed d* algorithm for real-time replanning. In: Proceedings of the international joint conference on artificial intelligence, pp 1652–1659
28. Stentz A, Mellon IC (1993) Optimal and efficient path planning for unknown and dynamic environments. Int J Robot Autom 10:89–100
29. Thrun S, Burgard W, Fox D (2005) Probabilistic robotics (intelligent robotics and autonomous agents). The MIT Press, Cambridge
30. Zanlungo F, Ikeda T, Kanda T (2011) Social force model with explicit collision prediction. EPL (Europhys Lett) 93(6):68,005. https://doi.org/10.1209/0295-5075/93/68005
31. Zenatti F, Fontanelli D, Palopoli L, Macii D, Nazemzadeh P (2016) Optimal placement of passive sensors for robot localisation. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4586–4593. https://doi.org/10.1109/IROS.2016.7759675
32. Zhang TY, Suen CY (1984) A fast parallel algorithm for thinning digital patterns. Commun ACM 27(3):236–239. https://doi.org/10.1145/357994.358023