Improving Throughput of Mobile Robots in Narrow Aisles

Simon G. Thomsen[1], Martin Davidsen[1], Lakshadeep Naik[2][a], Avgi Kollakidou[2][b], Leon Bodenhagen[2][c] and Norbert Krüger[2][d]

[1]*Faculty of Engineering, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark*

[2]*SDU Robotics, Maersk Mc-Kinney Mollar Institute (MMMI), Faculty of Engineering, University of Southern Denmark, Campusvej 55, Odense M, Denmark*

Keywords: Mobile Robotics, Human-Robot-Interaction.

Abstract: Emergency brakes applied by mobile robots to avoid collision with humans often block the traffic in narrow hallways. The ability to smoothly navigate in such environments can enable the deployment of robots in shared spaces with humans such as hospitals, cafeterias and so on. The standard navigation stacks used by these robots only use spatial information of the environment while planning its motion. In this work, we propose a predictive approach for handling dynamic objects such as humans. The use of this temporal information enables a mobile robot to predict collisions early enough and avoid the use of emergency brakes. We validated our approach in a real-world set-up at a busy university hallway. Our experiments show that the proposed approach results in fewer stops compared to the standard navigation stack only using spatial information.

1 INTRODUCTION

Today, Autonomous Mobile Robots (AMR) are widely used for logistic transportation in warehouses (Allied-Market-Research, 2019). They typically use a occupancy grid representation of the environment to localize themselves and navigate by first making a plan on a global level and then execute this global plan using a local planner. The global planner uses spatial heuristics such as shortest distance from point A to B for computing the path using search algorithms such as A* (Hart et al., 1968). The local planner uses the robot kinematic model (such as differential drive) to predict all possible trajectories for the specified look-ahead time in the future based on the robot's costmap and global plan, and selects the trajectory that doesn't involve a collision with any obstacles while trying to follow the global plan. Local planning is often accomplished using algorithms such as Dynamic Window Approach (DWA) (Fox et al., 1997), Elastic band planner (Rösmann et al., 2017) or Vector Field Histogram (VFH) (Borenstein and Koren, 1991).

Motivated by the success of these autonomous mobile robots in warehouses, many hospitals or other public institutions are trying to integrate mobile robots to perform logistic tasks into their everyday workflows (Fragapane et al., 2020), for example to reduce the non-nursing related workload (Yen et al., 2018). However, the structure as well as dynamics of these environments is quite different compared to the warehouse environments. Since robots often have to navigate through narrow hallways in the vicinity of humans. This can result in a significant number of emergency stops during navigation due to simplistic management of collisions in the standard navigation stack and can block the traffic in hospital hallways, which can be costly during emergencies. Because of that, some hospitals have even abandoned the use of mobile robots (DR, 2019).

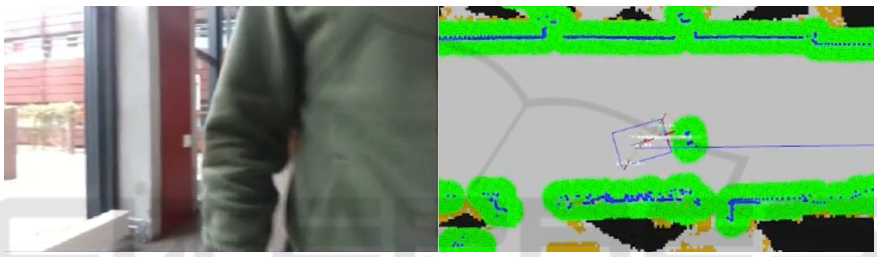
Let's take the example in Fig. 1 to understand the problem. In Fig. 1a, we see that the mobile robot (blue rectangle) is navigating across the hallway by following its global plan (blue line). A person (2 blue dots and green cost originating from the person's legs) starts approaching in opposite direction and we can see that it will result in a collision if the robot doesn't deviate from its global plan. As robot and person move towards each other, the cost associated with the person falls within the prediction horizon of the local planner and it starts to plan a local trajectory that will avoid a collision with the person (see Fig. 1b). However, before the robot can execute this planned trajectory, the person is already very close to the robot and finally the robot just stops in the mid-

[a] https://orcid.org/0000-0002-2614-8594
[b] https://orcid.org/0000-0002-0648-4478
[c] https://orcid.org/0000-0002-8083-0770
[d] https://orcid.org/0000-0002-3931-116X

Thomsen, S., Davidsen, M., Naik, L., Kollakidou, A., Bodenhagen, L. and Krüger, N.
Improving Throughput of Mobile Robots in Narrow Aisles.
DOI: 10.5220/0011717500003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 2: HUCAPP*, pages 221-228
ISBN: 978-989-758-634-7; ISSN: 2184-4321

(a) Person appears in the robots costmap.

(b) Person continue moving towards the robot.



(c) Robot starts to plan local trajectory to avoid collision with the person, but it is too late and the situation end in an emergency stop.

Figure 1: Handling of interaction with humans in the standard navigation system. Images to the left show the camera feed from the system; images to the right show visualized data from the system. The robot (blue rectangle) is following the global path (blue line). Green pixels indicate obstacles, while grey pixels indicate free space.

dle of the hallway to avoid a collision. One simple solution for this is to increase the look-ahead time of the local planner, however, this is computationally expensive as local planners are required to run at a very high frequency.

Recent improvements in deep learning have significantly improved the perception capabilities of the robots to detect and track humans etc. (Toshev and Szegedy, 2014; Mehta et al., 2017; Juel et al., 2020). This has made it possible to consider costs based on the context such as social groups to enable human-friendly trajectories (Charalampous et al., 2017; Kollakidou et al., 2021). However, in addition to this contextual information, humans also use temporal information while making navigation decisions such as how fast and in which direction someone is moving. Inspired by this navigation behaviour of humans, we implement a predictive way of handling dynamic objects and avoiding collisions. We show that by predicting the future trajectories of humans, we can reduce the number of times the robot has to stop in situations with high person densities and thus improve the throughput of mobile robots in narrow aisles.

2 RELATED WORK

In this section, we describe related work in human pose estimation, human motion prediction, predictive navigation and then summarize our contribution.

Human Pose Estimation and Tracking: Data-driven approaches (Wang et al., 2021) such as OpenPose (Cao et al., 2017) have made it possible to accurately track the different human joints in real-time on low-cost hardware. This has resulted in several new possibilities for robots such as task learning (Zimmermann et al., 2018), socially aware navigation (Yang et al., 2019) etc. Further, Juel et al. (Juel et al., 2020) have shown that these data-driven methods can be used in combination with probabilistic tracking

frameworks such as Kalman filter to track the 3D pose of the humans for use on mobile robots.

Human Motion Prediction for Robot Navigation: (Helbing and Molnar, 1995) have proposed a social force model for predicting human motion. Some other works have used well-engineered features related to humans or environments to learn the human motion using techniques such as inverse reinforcement learning (Henry et al., 2010), inverse optimal control (Kitani et al., 2012) etc. Recent works (Alahi et al., 2014; Alahi et al., 2016; Chen et al., 2019) have used experiences to learn and predict human motion. These are early recognition approaches, i.e. they monitor the motion for some time and then predict the object trajectory.

Predictive Navigation: (Chung and Huang, 2011) have proposed A* predictive motion planner to incorporate human motion while planning navigation using Dynamic Bayesian Networks. Thompson et al. (Thompson et al., 2009) have also presented a similar probabilistic motion model. (Unhelkar et al., 2015) have used anticipatory indicators of human motion to plan the robot's motion. Recent work by (Chen et al., 2019) have directly tried to learn to avoid collision with humans during navigation using an end to end approach.

In this work, our focus is to improve the commonly used ROS navigation stack (Guimarães et al., 2016) by reducing the number of emergency robot stops in narrow aisles. As shown by Helbind et al. (Helbing and Molnar, 1995), humans often tend to walk in hallways on the left or right lane. Thus, instead of using complex motion models for predicting human motion, we rely on real-time human pose estimators and use the tracking by detection paradigm similar to Juel et al. (Juel et al., 2020) to track and predict human motion using probabilistic Kalman filter. We also show that associating information from the robot's LiDAR and camera results in better tracking performance. Instead of directly planning the robot motion on the human motion (Chen et al., 2019; Unhelkar et al., 2015; Thompson et al., 2009), we exploit predictions based on a Kalman filter that enable the robot to modigy its local plan early enough enabling a smooth robot motion in the vicinity of humans and reducing the number of unintended stops. Our solution can easily be integrated with the standard navigation stacks (Guimarães et al., 2016) used by many mobile robots today.

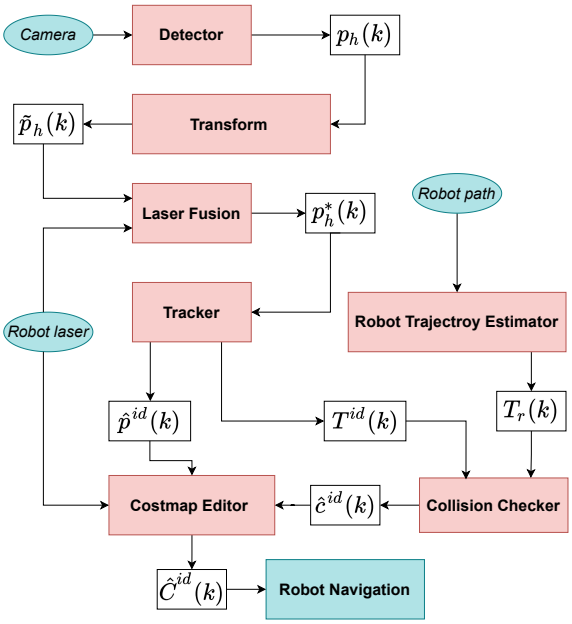


Figure 2: System overview - components colored in blue are part of the mobile robot navigation stack, components colored in red are introduced in the proposed solution.

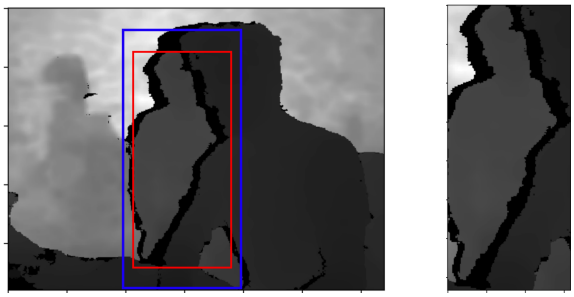
3 METHODOLOGY

Fig. 2 depicts an overview of our system. The *Detector* finds a human h and its corresponding 3D coordinates from camera data in frame k. The 3D position of the human in the camera frame is indicated as $p_h(k)$. These 3D coordinates are then transformed (see *Transform*) into the map frame as $\tilde{p}_h(k)$ and then merged with the laser data for stabilization purposes, creating a 3D position $p_h^*(k)$ in the map frame.

The *Tracker* associates 3D detections across the image sequence arriving at tracks $\hat{p}^{id}(k)$ where id indicates the same person across time. Hereafter, the prediction of the to be expected trajectory of the tracked human $T^{id}(k)$ is computed and is passed to the *Collision Checker*.

These tracks need to be compared with the trajectory of the robot $T_r(k)$. The collision checking is then performed to check for possible intersections $\hat{c}^{id}(k)$ between the robot's trajectory and the predicted trajectory of each human.

The *Costmap Editor* manipulates the costmap $\hat{C}^{id}(k)$ to allow for the robot to navigate taking the predicted positions of humans into account. In the following subsections, we describe these components in detail.



(a) (b)

Figure 3: A) shows a captured depth image with the bounding box (blue) for the detected human and the downscaled bounding box (red). b) shows the cropped image used for human position estimation.

3.1 Human Detection

Human detection involves the computation of 2D pixel coordinates $p_{2,h}(k)$ for different humans h for the k-th image and transforming the respective 3D positions $\tilde{p}_h(k)$ to the map frame.

We use CenterNet (Zhou et al., 2019) for human detection. The output from image k is the center of the detected human $p_{2,h}(k)$ and a corresponding bounding box $S_h(k)$. To distinguish between robot movement and human movement, the center is transformed to the robot's map frame. This transformation is performed as follows: First, the distance $d_h(k)$ to human h is obtained through the depth image. Next, the bounding box estimate $S_h(k)$ is used to crop the depth image to fit the object of interest. To further ensure that mainly the desired object is captured, the size of the bounding box is reduced by 20% as seen in Fig. 3a. The resulting depth information can be seen in Fig. 3b.

In Fig. 3 it can be seen that humans may be occluded and that multiple areas of the image could correspond to the human in question. We apply K-means clustering on the bounding box to determine the most likely distance $d_h(k)$ of the human to the camera from the cropped depth image by identifying the largest cluster.

This distance is then used to project the 2D center pixel coordinate, $p_{2,h}(k)$ into the 3D coordinate $p_h(k)$. This is done by finding the unit vector passing through the camera center to the pixel coordinate and extending the vector with the found distance. Finally, the transformation between the coordinate frames is used to convert the 3D position to the map frame ($\tilde{p}_h(k)$).

When the robot moves, the variance of the estimated human position increases due to the limited frame rate of the camera and the rolling shutter of the RGB sensor. Furthermore, under rotational movement, static objects are erroneously tracked as moving, due to a communication delay between camera and robot. This makes it necessary to improve the estimate of the positions as done in the following section.

3.2 Camera and Laser Sensor Fusion

Since the laser sensor of the robot is much more precise than the 3D positions $\tilde{p}_h(k)$ computed from the camera, we perform fusion of the laser and camera data and arrive at improved estimates $p_h^*(k)$.

A fusion algorithm is implemented based on creating a K-dimensional tree of the laser scan point cloud $z(k)$ for efficient nearest neighbor range searches. The search point used is the human detection $\tilde{p}_h(k)$ result in associated points $z_h(k)$. The impact of the fusion algorithm can be seen in Fig. 4.

The centroid of the associated laser data $z_h(k)$ provides an estimate of the position of the human $\bar{z}_h(k)$. The centroid is given by the mean of associated laser scan points. This is then fused with the human detection with bias λ to give different weights to the human detection $\tilde{p}_h(k)$ and the centroid $\bar{z}_h(k)$. The resulting estimated human position $p_h^*(k)$ is given by (1).

$$p_h^*(k) = \lambda \bar{z}_h(k) + (1-\lambda)\tilde{p}_h(k) \quad (1)$$

where we used $\lambda = 0.3$.

3.3 Human Tracking

The information computed in section 3.2 is still for a single frame. In this subsection, we connect these individual estimates to tracks across different image frames in which the human track position is represented by $\hat{p}^{id}(k)$ for a given track id at time k. From this, we can compute the velocity and perform predictions about the future state of the human (as done in section 3.4).

Data Association. To be able to use velocity information of humans, the DeepSORT (Wojke et al., 2017) object tracker is used. We use the Kalman filter and the Hungarian method (Kuhn, 1955) with an association metric for frame-by-frame data association that combine both motion and appearance information. The original DeepSORT algorithm, tracks in the image plane, which only provides 2D information about found objects. Hence the tracker is modified to track in the 3D map frame according to (Juel et al., 2020).

The Kalman filter model is chosen according to the dynamics of humans. Here a constant velocity Kalman filter is used, which thereof assumes nearly constant velocity.

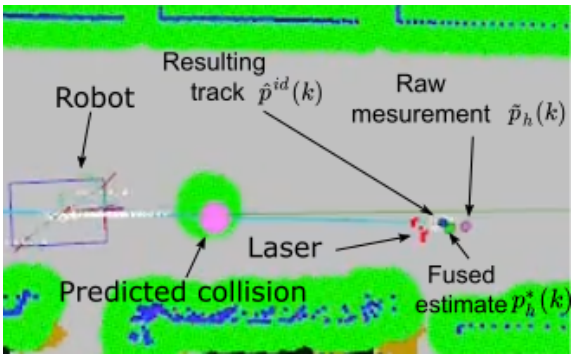


Figure 4: Camera and laser data fusion for improved human pose estimation. The laser data points (red points), the raw measurement $\tilde{p}_h(k)$ from the human detection (purple), the measurement from the fused estimate $p_h^*(k)$ (green) and the resulting track influenced by the Kalman filter $\hat{p}^{id}(k)$ (blue).

The motion information from the Kalman filter is incorporated by calculating the squared Mahalanobis distance between measurements and predicted Kalman states of tracks. However, the squared Mahalanobis distance favors tracks with larger uncertainties. Therefore, a matching cascade that matches the detections and tracks is set up to prioritize tracks with smaller ages.

The appearance information is implemented by computing an appearance descriptor for each bounding box. The last descriptors are then stored in a gallery for each track. The smallest cosine distance between track and detection appearances is finally computed. The resulting track $\hat{p}^{id}(k)$ and improvement hereof can be seen in Fig. 4.

Track Maintenance. To handle incoming, persistent and outgoing humans with respect to the field of view of the camera, track maintenance is required. Tracks are initiated for each detection that cannot be associated with an existing track and are classified as tentative. Tentative tracks are expected to have a successful measurement to track association for n consecutive frames to be classified as confirmed tracks. If no association occurs or the track age reaches the set max-age the track is deleted.

3.4 Predicting Object Trajectories

To be able to predict potential collisions, as done in Section 3.5, the future position of the human needs to be computed. For that, we predict the likely trajectory $T^{id}(k)$ of the tracked human $\hat{p}^{id}(k)$ up to five seconds ahead from the current state of the track.

The Kalman filter can create multiple predictions, from the current state and covariance for each tracked human. A range of timestamps is defined and used for prediction, which thereby results in a trajectory prediction for each track.

$$T^{id}(k) = \{\hat{p}^{id}(k+0|k), \hat{p}^{id}(k+0.5|k), ..., \hat{p}^{id}(k+5|k)\} \quad (2)$$

where $\hat{p}^{id}(k+i|k)$ is the predicted position of track id id at time $k+i$ based on current state k computed for every half second up to 5 seconds in the future.

3.5 Collision Detection

To modify the costmap, as done in Section 3.6, a predictive collision system is necessary. Based on the predicted path $T^{id}(k)$, we can compute whether collisions occur on the planned path of the robot. A collision $\hat{c}^{id}(k)$ occurs when the human trajectory $T^{id}(k)$ intersect the robot trajectory $T_r(k)$ at the approximate same location and time.

As standard, the robot only provides a path, which doesn't contain time information. Therefore, a trajectory must be computed before the collision checking can be performed. The acceleration of the robot varies during the execution making the exact trajectory unknown. Instead, an approximation is used as the robot shares information about its desired velocity. The approximate velocity throughout the path is then estimated to be the mean of the current velocity and desired velocity.

Collision checking is performed by checking if the Euclidean distance between any point from the predicted trajectories of tracks $T^{id}(k)$ and any point along the trajectory of the robot $T_r(k)$ is within a defined distance threshold and at a time difference lower than the set time threshold. If both constraints are accepted, a possible collision $\hat{c}^{id}(k)$ is found.

3.6 Costmap Editor

Based on the predictions of the human, the costmap is modified to take the changed dynamic situation into account: Given the estimated position $\hat{p}^{id}(k)$, we define a circle around $\hat{c}^{id}(k)$ with radius r marking the potential collision and by that increasing the respective costs in the cost map to avoid that the robot is planning a path through that area.

Further, we perform a nearest neighbour range search on the estimated position $\hat{p}^{id}(k)$ from which the human has moved away decreasing the costs in the costmap in that respective area in a certain radius r.

Currently, when the robot computes a new global path, it uses the combination of the local and global costmap containing recorded laser data where found obstacles are inflated for safer maneuvering. Hence a sub-optimal plan is often the result in a dynamic

environment. The costmap editing is performed to remove the laser data for humans which are tracked and by that the planner is able to plan based on what is to be expected and not the originally recorded situation, allowing the robot to generate a more optimal path. The improvement of the proposed system can be seen in Fig. 5.

For stationary or intersecting humans, the robot has information of the potential collisions from the collision detection described in Section 3.5. Furthermore, in the event of imprecise tracking or other possible failures, the original collision checking system is still based on the unmodified laser data and applied to ensure safe operation. As a consequence, the robot would drive around the person in due time in each of these cases.

4 EXPERIMENTAL EVALUATION

Experimental Set-Up. To validate the performance of the proposed system over the original system, both systems are tested using the setup illustrated on Fig. 6. An external camera counting the people walking through the area operates at a rate of 0.5 Hz.

The mission of the robot was to repeatedly drive between two points placed 24 m apart at the side of the aisle (see Fig. 6). The testing period expands over 4 days, consisting of one early and one late time slot, both within the peak time at around 12pm, with a duration of 20 min. The peak time was chosen as experiments showed that people tend to avoid the robot in light densities, with such margin that the system would have too little influence to indicate a different behavior. Throughout the test period, the robot was equipped with either the original system without predicting collisions or the proposed system for the early time slot and the other system for the late time slot. Each day the system order was flipped. The people walking in the aisle are not aware of the purpose of the test and are assumed to be unbiased towards the performance of the robot.

Results. The collected metrics to be compared (see table 1) are the number of stops, the average number of stops per run, average duration timed per run, the average velocity computed per run, the average population size per min and the average number of stops per person in the aisle. One run hereby means from point A to B or vice versa. The average number of stops per person in the aisle will give a comparable metric as the average population size and the number of stops are correlated. Hence this will take the population size difference at different runs out of the equation.

Table 1: Collected data from 4 days of testing using the original and the proposed system.

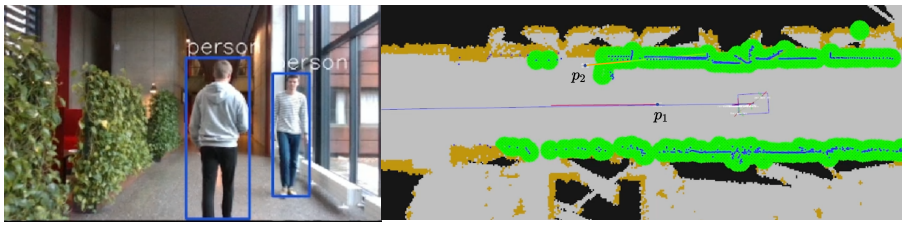
	Day 1	Day 2	Day 3	Day 4	Overall
Number of stops					
Original	19	10	5	8	42
Proposed	7	10	6	12	35
Average stops per run					
Original	0.73	0.39	0.19	0.29	0.39
Proposed	0.30	0.39	0.25	0.44	0.35
Average duration per run [s]					
Original	43.65	42.27	40.22	39.67	41.45
Proposed	41.30	42.06	41.29	42.73	41.88
Average velocity [m/s]					
Original	0.58	0.59	0.61	0.60	0.60
Proposed	0.59	0.59	0.60	0.58	0.59
Average population size per min					
Original	9.85	8.5	5	11.55	8.73
Proposed	12.4	9.75	11.95	11.3	11.35

To avoid repeatedly counting stops, a stop has been defined as follows. The distance from the robot to the goal has to be bigger than 1 m to avoid counting natural stops when turning. Furthermore, two thresholds are used to ensure that the robot starts moving again before counting a new stop. Hence the robot has to reach the upper threshold of $0.3\,\mathrm{m/s}$ and then slow down under the lower threshold of $0.05\,\mathrm{m/s}$. The accomplishment of reaching a threshold will be reset for each run. The lower threshold is not set to $0\,\mathrm{m/s}$ because of practical reasons such as the risk of not counting a stop, as the published velocity never stabilizes at exactly $0\,\mathrm{m/s}$. The results from the testing can be seen on Table 1.

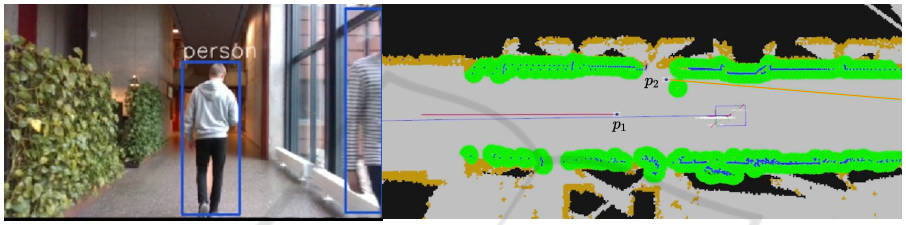
The results show an increase in performance with regards to the number of stops and an average number of stops per run (first two rows in table 1). The number of stops decreased with 17 % although our system in average has dealt with more difficult situations (i.e., a higher average population).

Table 1 also shows a very slight decrease in the overall performance of the average duration per run and average velocity (third and fourth row). Practically, avoiding a stop should help increase the performance of these metrics. A possible reason for this is the addition of the collision system combined with the behavior of the navigation stack of the MiR robot: When the robot is in the proximity area of an obstacle, the navigation stack will automatically slow down the robot for safer maneuvering. Hence the probabilistic collision checking can cause slowing down for colli-

(a) Before generated path.



(b) After generated path.



(c) Following the path.

Figure 5: A) The environment before the robot generates a path. Person p_1 is directly in front and moving away from the robot, while person p_2 is moving towards the robot. Note that at that point no reliable trajectories for the movement of the two persons has been computed yet. b) The generated global path by the robot, which goes directly through p_1. c) The robot is following the computed path because it has taken into account that person p_1 has moved and hence it is possible to continue without colliding with p_1.

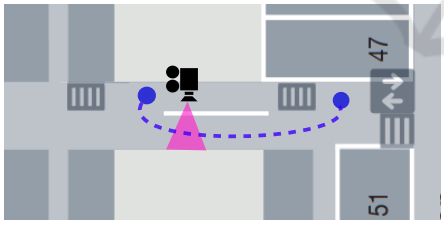


Figure 6: The test setup: The purple area represent the field of view of the camera for people counting; the dashed line suggest a possible path for the robot from start to end point.

sions that are not going to happen, making it a possible side effect of the proposed system.

5 CONCLUSIONS

In this work, we have investigated the problem of emergency stops applied by mobile robots while navigating in narrow aisles in the vicinity of the humans. We therefore proposed a predictive navigation approach that predicts the collision with humans early enough to adapt the trajectory and to avoid the use of emergency brakes. Our results indicate a reduction of the number of stops compared to the standard navigation stack. However, more testing needs to be performed to further substantiate the results as well as more development work to achieve a smoother integration of our approach in the MiR softare architecture.

ACKNOWLEDGEMENTS

This work is supported by the Innovation Fund Denmark for the project DIREC (9142-00001B).

REFERENCES

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. (2016). Social lstm: Human trajectory prediction in crowded spaces. In *IEEE conference on computer vision and pattern recognition*, pages 961–971.

Alahi, A., Ramanathan, V., and Fei-Fei, L. (2014). Socially-aware large-scale crowd forecasting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2203–2210.

Allied-Market-Research (2019). Mobile logistic robot - market size and industry.

Borenstein, J. and Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Real-time multi-person 2d pose estimation using part affinity fields. In *IEEE conference on computer vision and pattern recognition*, pages 7291–7299.

Charalampous, K., Kostavelis, I., and Gasteratos, A. (2017). Recent trends in social aware robot navigation: A survey. *Robotics and Autonomous Systems*, 93:85–104.

Chen, C., Liu, Y., Kreiss, S., and Alahi, A. (2019). Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *International Conference on Robotics and Automation*, pages 6015–6022. IEEE.

Chung, S.-Y. and Huang, H.-P. (2011). Predictive navigation by understanding human motion patterns. *International Journal of Advanced Robotic Systems*, 8(1):3.

DR (2019). Hospital sætter robotter for millioner i garagen: Kunne ikke færdes blandt mennesker.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33.

Fragapane, G., Hvolby, H.-H., Sgarbossa, F., and Strandhagen, J. O. (2020). Autonomous mobile robots in hospital logistics. In *IFIP International Conference on Advances in Production Management Systems*, pages 672–679. Springer.

Guimarães, R. L., Oliveira, A. S. d., Fabro, J. A., Becker, T., and Brenner, V. A. (2016). Ros navigation: Concepts and tutorial. In *Robot Operating System (ROS)*, pages 121–160. Springer.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.

Helbing, D. and Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282.

Henry, P., Vollmer, C., Ferris, B., and Fox, D. (2010). Learning to navigate through crowded environments. In *IEEE International Conference on Robotics and Automation*, pages 981–986. IEEE.

Juel, W. K., Haarslev, F., Krüger, N., and Bodenhagen, L. (2020). An integrated object detection and tracking framework for mobile robots. In *International Conference on Informatics in Control, Automation and Robotics*, pages 513–520. SCITEPRESS Digital Library.

Kitani, K. M., Ziebart, B. D., Bagnell, J. A., and Hebert, M. (2012). Activity forecasting. In *European conference on computer vision*, pages 201–214. Springer.

Kollakidou, A., Naik, L., Palinko, O., and Bodenhagen, L. (2021). Enabling robots to adhere to social norms by detecting f-formations. In *IEEE International Conference on Robot & Human Interactive Communication*, pages 110–116. IEEE.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.-P., Xu, W., Casas, D., and Theobalt, C. (2017). VNect: Real-time 3D human pose estimation with a single RGB camera. *ACM Transactions on Graphics (TOG)*, 36(4):1–14.

Rösmann, C., Hoffmann, F., and Bertram, T. (2017). Kinodynamic trajectory optimization and control for car-like robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5681–5686. IEEE.

Thompson, S., Horiuchi, T., and Kagami, S. (2009). A probabilistic model of human motion and navigation intent for mobile robot path planning. In *International Conference on Autonomous Robots and Agents*, pages 663–668. IEEE.

Toshev, A. and Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *IEEE conference on computer vision and pattern recognition*, pages 1653–1660.

Unhelkar, V. V., Pérez-D'Arpino, C., Stirling, L., and Shah, J. A. (2015). Human-robot co-navigation using anticipatory indicators of human walking motion. In *IEEE International Conference on Robotics and Automation*, pages 6183–6190. IEEE.

Wang, J., Tan, S., Zhen, X., Xu, S., Zheng, F., He, Z., and Shao, L. (2021). Deep 3d human pose estimation: A review. *Computer Vision and Image Understanding*, 210:103225.

Wojke, N., Bewley, A., and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric.

Yang, C.-T., Zhang, T., Chen, L.-P., and Fu, L.-C. (2019). Socially-aware navigation of omnidirectional mobile robot with extended social force model in multi-human environment. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1963–1968. IEEE.

Yen, P.-Y., Kellye, M., Lopetegui, M., Saha, A., Loversidge, J., Chipps, E. M., Gallagher-Ford, L., and Buck, J. (2018). Nurses' time allocation and multitasking of nursing activities: a time motion study. In *AMIA Annual Symposium*, volume 2018, page 1137. American Medical Informatics Association.

Zhou, X., Wang, D., and Krähenbühl, P. (2019). Objects as points.

Zimmermann, C., Welschehold, T., Dornhege, C., Burgard, W., and Brox, T. (2018). 3D human pose estimation in RGBD images for robotic task learning. In *IEEE International Conference on Robotics and Automation*, pages 1986–1992. IEEE.