

Probabilistic Mapping of Dynamic Obstacles Using Markov Chains for Replanning in Dynamic Environments

Florian Rohrmüller, Matthias Althoff, Dirk Wollherr and Martin Buss

Abstract—Robots acting in populated environments must be capable of safe but also time efficient navigation. Trying to completely avoid regions resulting from worst case predictions of the obstacle dynamics may leave no free space for a robot to move, especially in environments with high dynamic. This work presents an algorithm for a "soft" risk mapping of dynamic objects leaving the complete space free of static objects for path planning. Markov Chains are used to model the dynamics of moving persons and predict their potential future locations. These occlusion estimations are mapped into risk regions which serve to plan a path through potentially obstructed space searching for the trade-off between detour and time delay. The offline computation of the Markov Chain model keeps the computational effort low, making the approach suitable for online applications.

I. INTRODUCTION

A major goal in current mobile robotics research is to bring robots into natural human-populated environments. This imposes great challenges on human-robot interaction capabilities. Action in human environments and interaction with humans is one of the striving goals of the Autonomous City Explorer (ACE) project [1]. The objective of the project is to create a robot (Fig. 1) which explores an unstructured and populated urban environment, given neither map knowledge nor being equipped with GPS. Instead it is primarily gathering information through interaction with people. One fundamental task within this scope is a sophisticated navigation in densely populated environments.

Such scenario requires safe navigation amongst people. However, too passive and conservative motion may not work in densely populated places. To be able to move deftly and goal-oriented within populated areas, the environmental dynamics has to be predicted and taken into account during path planning. This requires first to perceive dynamic objects, second to predict their future motions and third to plan a sophisticated path based on this knowledge. The focus of this paper is primarily on the second step, in presenting an algorithm for a "soft" mapping of predicted dynamics into risk areas. Instead of "hard" avoidance of obstacle areas, the perceptual reasoning system provides a mean to plan through potentially obstructed space searching for the trade-off between detour and time delay.

The remainder of the paper is structured as follows: The next section provides an overview about related work. Section III starts with an outline of the overall system in which

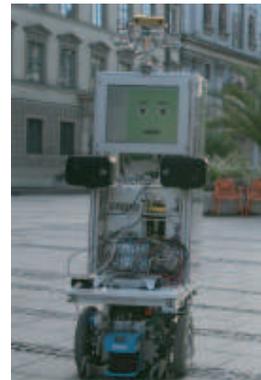


Fig. 1. The Autonomous City Explorer (ACE) robotic platform.

the presented approach is embedded, followed by a detailed description of the risk map computation and how these can be used for replanning. Section IV gives experimental results followed by a conclusion in Section V.

II. RELATED WORK

The capability to distinguish between the static environment and moving persons requires a people tracking. Recognition of people has been widely studied using vision systems, e.g. [2], [3], [4]. While stereo vision systems like [3], [4] provide an expedient way to retrieve depth information from a camera scene, visual systems are facing problems when being confronted with varying lightning conditions. Consequently they are not the best choice for outdoor scenarios. An alternative way to perceive and track people is using laser range finders as in [5] and [6], having the advantage of robustness to sunlight. Furthermore, the gathered depth information suffers usually less noise and covers distances far beyond reliable visual depth reconstruction.

In addition to people tracking, a major challenge of navigation within populated environments is considering the perceived people during path planning. While safety can be ensured by using a low level obstacle avoidance like [7], path planning focusing on smooth continuous motion demands an efficient handling of the environmental dynamics.

A lot of research is engaged in the path finding problem of robots navigating in dynamic environments. In [8] the Multi-partite Rapidly-exploring Random Tree (MP-RRT) algorithm for efficient path planning in dynamic environments is introduced. The single-query character of RRTs is extended by subtree re-use and planning around moving obstacles with unknown dynamics. A combination of probabilistic path planning with deterministic approaches – using Probabilistic Roadmaps (PRMs) and Anytime D* [9] respectively – is presented in [10]. This approach provides anytime replanning

in situations where an agent may have no full information about the dynamics of the environment. Besides the path length, further cost metrics such as traversal risk and time, stealth or visibility are mentioned. Dynamic obstacles are modeled as space-time volumes using a deterministic or worst case approach. In [11] Cell-based PRMs (CPRMs) are presented, which focus on efficient replanning especially in large environments. In [12] the velocity obstacle (VO) is introduced, being the conical velocity space derived from the set of colliding relative velocities between robot and moving object. Under the assumption of constant velocities, any relative velocity within the VO is causing a collision. The VO is extended in [13] by a probabilistic version (PVO), where the perception system is modeled probabilistically while the obstacle model itself remains deterministic.

Modeling of dynamic obstacles is also extensively covered in the field of computer graphics. In [14] a neighborhood graph is used for collision detection in a crowd of virtual humans. The collision avoidance is based on linear trajectory extrapolation and local speed adaptation for each virtual human. In [15], Finite State Machines (FSM) are used to model virtual humans. While dealing with virtual humans provides to use deterministic dynamics of the persons, these approaches usually do not cover all circumstances facing a robot in the real world. Consequently they are often leading to inaccurate results when applied there.

The above mentioned approaches have in common, that they aim for collision free path planning, i.e. hard avoidance of potentially obstructed space. This can be a necessity in many scenarios, thinking e.g. about applications with high velocities like autonomous cars [16] where the acceptable collision rate drops down to zero. On the other hand, trying to completely avoid intersections with any potential obstacle trajectory may leave no feasible path at all, especially within densely populated environments.

In [17] a probabilistic navigation function (PNF) is presented, using a probabilistic risk map for path planning with E^* [18]. The risk map is derived from a co-occurrence estimation of dynamic obstacle and robot, which is based on worst-case assumptions assuming that the object can change its speed (within a bounded interval) and direction of travel. In [19] motion patterns of people are learned to predict their future behaviors. In a pre-processing phase trajectories are derived from laser range data and clustered by using expectation maximization. The outcome then serves to learn a Hidden Markov Model (HMM) for each person, which later is used to predict the most likely trajectories of the tracked persons. This technique can perform very well for known environments and is able to keep track of persons even after they left the sensory range, at least for some time. However, a required pre-learning and a finite set of preferably well distinguishable trajectories make this approach not applicable in unknown environments with arbitrary trajectory runs.

Focused especially on unknown and densely populated environments, this work presents an approach using Markov Chain (MC) models to predict probabilistic reachable sets of persons, which are then used to compute a risk map

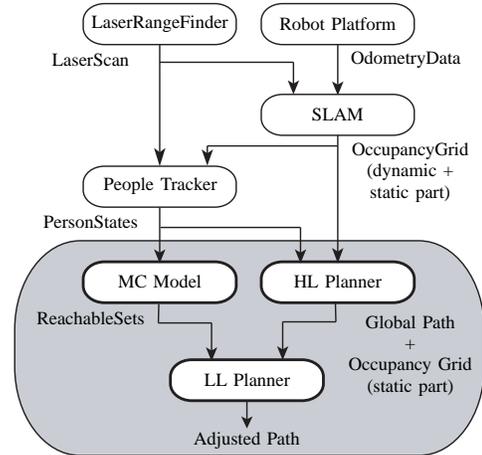


Fig. 2. Outline of the proposed approach.

for replanning. MCs, a probabilistic version of FSMs, are in [20] and [21] already successfully applied to verify safe locomotion of cognitive cars and are here used to model the dynamics of real humans. Using risk regions instead of hard dynamic obstacle avoidance provides a soft risk measure while leaving the complete space free of static objects for path planning. Provided that a reliable low-level collision avoidance prevents real collisions, the worst-case is a time delay resulting in a situation where the robot has to stop.

In the next section is described how these reachable sets are derived from raw laser range data and then mapped into risk regions. Furthermore a cost metric is presented to determine the trade-off between a detour and a time delay during path planning.

III. APPROACH

This section starts with an outline of the overall system followed by a description of its submodules, where the focus is on the risk map computation and how these can be used for replanning.

A. Algorithm Overview

Several subtasks are necessary to perceive dynamics in the environment, predict its future changes and plan the most promising path with respect to it. Fig. 2 gives an overview of the entire system in which the risk mapping is embedded.

Laser range and odometry data serve as input to create and update an occupancy grid by applying grid-based Simultaneous Localization and Mapping (SLAM).

The people tracker matches the occupancy grid with the laser scans to extract moving persons and provide an estimate of their current states consisting of positions and velocities.

A high level (HL) path planner removes the objects associated with the person states from the occupancy grid. Then the remaining static part of the occupancy grid is used to build a visibility graph on which an A^* search is performed, resulting in an optimal static global path [22].

In parallel, a Markov Chain (MC) model predicts the probabilistic reachable sets for each person using the state estimations retrieved from the people tracker, as described

below. The reachable sets provide the probabilities of the person dynamics being in a certain state within a certain time interval.

A low level (LL) path planner transforms the probabilistic reachable sets from the MC model into risk maps, which provide the probability that the corresponding person will occupy a certain grid cell in a certain time interval. Then a greedy path search is applied using a cost function incorporating static path costs and the risk maps to determine the time-minimizing compromise between path length and travel speed.

B. SLAM and People Tracker

The SLAM module and the People Tracker have been implemented as described in [6]. For SLAM, a Rao-Blackwellized particle filter and an Iterative Closest Point approach is used in combination with odometry information. The people tracker combines conditional particle filters [5] with SLAM for enhanced data association. Scan matching of the current with previous measurements is performed to perceive moving persons and provide for each perceived person an estimate of its current state vector

$$\hat{P} = (x, y, v_x, v_y)^T, \quad x, y, v_x, v_y \in \mathbb{R}, \quad (1)$$

where x and y are the x - and y - position and v_x and v_y are the velocities in x and y direction in global coordinates.

C. High Level Path Planning

The high level path planner is used to extract the static parts of the occupancy grid and plan a path from the current robot location to a given goalpoint within it. The goalpoint is retrieved through the behavior selection described in [6]. Depending on the current situation, the robot decides whether to explore the environment, approach a person, drive to a specific location or improve SLAM via loop closing.

To reach the next goalpoint, the robot has to navigate outdoors in unknown, dynamic, and unstructured environment. Therefore frequent and complete replanning is required. Methods like Voronoi Diagrams or Potential Fields are rather inappropriate for this purpose, since they either might result in huge detours by trying to keep the robot as far away from obstacles as possible or they suffer from high computational costs. Therefore a method suitable for outdoor scenarios [22] is applied, using a visibility graph on which A*-search is performed. Fig. 3 shows an occupancy grid with visibility graph and static path, derived online with the ACE robot.

In a first step, all frontiers and obstacles within the occupancy grid are determined. Frontiers are connected unexplored cells having each at least one free neighboring cell, while obstacles are defined as connected occupied cells. In a next step the position $(x, y)_\pi$ of each perceived person π is transformed into cell coordinates $c_\pi = \langle m, n \rangle_\pi$ ($0 \leq m < w_{og}$; $0 \leq n < h_{og}$; $m, n, w_{og}, h_{og} \in \mathbb{N}_0^+$), where w_{og} and h_{og} are the number of columns and rows of the occupancy grid respectively. Then each obstacle o_i for which holds $c_\pi \in o_i$ is removed from the grid by setting the value of all cells $c_{o_i} \in o_i$ to free space, leaving only the static obstacles for the HL path planning.

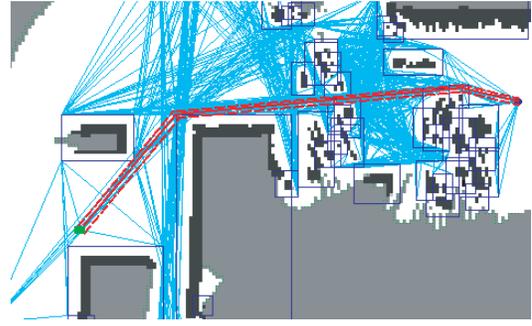


Fig. 3. Occupancy grid with visibility graph (solid cyan lines) and static path (dashed red lines), which is blown up by the width of the robot (green).

Within this static grid, obstacles are clustered into rectangular boxes on which C-Space projection is performed by blowing them up by the robot width, shown as blue rectangles in Fig. 3. The vertices of these rectangles serve as nodes to build a visibility graph and determine a global path by applying classical A* following [22]. This path is guaranteed to avoid static objects in an optimal manner and can be adjusted considering the dynamic parts as described in the following.

D. Human Markov Chain Model

In order to obtain time varying and probabilistic occupancy grids (risk maps) of humans surrounding the robot, the continuous dynamics of humans is abstracted to Markov chains. Two different models describing the dynamics of humans are considered. The first model, referred to as *velocity model*, assumes that the humans move at constant speed v with heading angle β . The resulting dynamic model for the x - and y -position is

$$\dot{x} = \cos(\beta)v, \quad \dot{y} = \sin(\beta)v. \quad (2)$$

The tracker data is used to initialize $v(0) = |(v_x, v_y)^T|$ and $\beta(0) = \angle(v_x, v_y)$.

The second model, referred to as *acceleration model*, is defined as a hybrid system. Hybrid systems combine continuous and discrete dynamics such that the system state is a combination of continuous and discrete states [23]. Discrete states are often referred to as locations or modes. The hybrid model of the humans has three modes: *standstill*, *acceleration* and *maximum velocity*. The system evolves in mode *standstill* if the velocity v is zero and in mode *maximum velocity* if v has reached maximum velocity v_{max} , which can be freely chosen. Otherwise, the mode *acceleration* is active which also considers deceleration if the acceleration has negative sign. The dynamics of the mode *standstill* is $\dot{x} = 0, \dot{y} = 0$ and of the mode *maximum velocity* is $\dot{x} = \cos(\beta)v_{max}, \dot{y} = \sin(\beta)v_{max}$. For the *acceleration* mode, the interval of possible acceleration values $a \in [\underline{a}, \bar{a}]$ ($\underline{a}, \bar{a} \in \mathbb{R}$) is introduced, such that the continuous dynamics of (2) is extended by

$$\dot{v} = a, \quad a \in [\underline{a}, \bar{a}], \quad (3)$$

where the interval $a \in [\underline{a}, \bar{a}]$ can be freely chosen. The abstraction from continuous dynamics to Markov chains is performed as presented in [20], [21]. Before applying this scheme, the state space and input space of (2) and (3) are

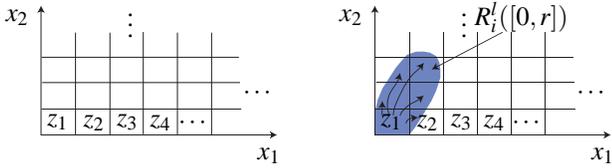


Fig. 4. Rectangularly discretized state space (left) and reachable set starting from a cell of the discretized state space (right).

gridded into rectangular cells of equal size, see Fig. 4 for an exemplary two dimensional state space.

The cells of the discretized state space are denoted z_q and the ones of the input space \hat{z}_p . The length ρ_z of the cells z_q in the dimension of the x - and y -position are chosen to be a natural multiple of the cell size ρ_c of the occupancy grid. This simplifies the matching of cells z_q of the reachable sets to corresponding occupancy grid cells c . The discretization of the velocity v in (3) and the inputs can be chosen arbitrary.

The discretization allows the abstraction of continuous space to discrete space, where each cell represents a discrete state and a region of the continuous space. The discrete state is denoted $q \in \mathbb{N}^+$ and the discrete input $\hat{q} \in \mathbb{N}^+$. The probability that the discrete state q has a certain value i is denoted $p_i = P(q = i)$, where $P(\cdot)$ is the operator returning the probability of an event and p is referred to as the probability vector. The probability vector for the time step $k+1$ ($k \in \mathbb{N}^+$ with time increment $\Delta t = r$) is denoted $p((k+1)r)$ and the one for a time interval $t \in [kr, (k+1)r]$ is denoted $p([kr, (k+1)r])$.

In a next step, the conditional probability that the system starting from state $q = i$ with input $\hat{q} = l$, is in state s within the time interval $t \in [0, r]$ is computed. These conditional probabilities $P(q' = s | q = i, \hat{q} = l, t \in [0, r])$ are stored in l matrices Φ_{si}^l where s is the row and i the column number. The conditional probabilities are obtained by computing the reachable set of the continuous system. The exact reachable set $R^e(r)$ of a system $\dot{x} = f(x) + v(t)$ where the uncertain input is taken from a bounded set $v(t) \in V \subseteq \mathbb{R}^n$ at time $t = r \in \mathbb{R}^+$ is defined as the set of states, that can be reached starting from a set of initial states X_0 under all possible uncertain inputs: $R^e(r) = \{x | x(t)$ is a solution of $\dot{x} = f(x) + v(t), t = r, x(0) \in X_0, v \in V\}$. Overapproximations of reachable sets are denoted $R(r) \supset R^e(r)$ and the overapproximated set for the time interval $t \in [0, r]$ is defined as the union of all $R(t)$ within $t \in [0, r]$: $R([0, r]) := \bigcup_{t \in [0, r]} R(t)$. Reachable sets that are computed from an initial set $X_0 = z_i$ are denoted $R_i^l([0, r])$ (indexed by the starting cell as well as the input cell). Under the assumption, that trajectories are equally distributed within $R_i^l([0, r])$, the probability of reaching the discrete state s is obtained geometrically by computing the intersection of the reachable set $R_i^l([0, r])$ with other cells z_s :

$$\Phi_{si}^l([0, r]) = \frac{V((R_i^l([0, r]) \cap z_s))}{V(R_i^l([0, r]))} \quad (4)$$

The operator $V(\cdot)$ returns the volume of a geometric object and the sets $R_i^l([0, r])$ are computed as presented in [24]. An illustration of this approach is given in Fig. 4 on the right. This procedure is repeated in an offline computation for all possible combinations of cells i, s, l . The left side in Fig. 5 shows a numerical example of an arbitrary reachable set

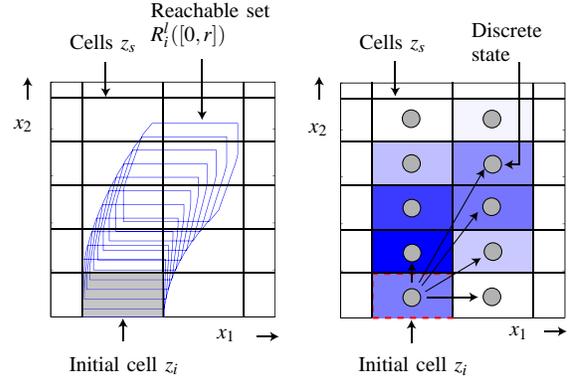


Fig. 5. Exemplary reachable set for $t = [0, r]$ (left) and corresponding probabilistic reachable set (right).

starting from $X_0 = z_i$. The right side shows the corresponding probabilistic reachable set from the same initial state obtained by the computed transition probabilities $\Phi_{si}^l([0, r])$. The darker the color of the reached cells, the higher is the probability of reaching that cell.

Besides the computation of the transition matrix $\Phi_{si}^l([0, r])$, the special case of a transition matrix $\Phi_{si}^l(r)$ for the time point $t = r$ is computed analogously. This allows to compute the probabilistic reachable set for both, time points and time intervals. This is done as the reachable set for time intervals is based on initial cells at the time point $t = 0$ such that the probability vector for the time interval is based on the probability vector for a time point:

$$\begin{aligned} p_s((k+1)r) &= \Phi_{si}^l(r) p_i(kr) \\ p_s([kr, (k+1)r]) &= \Phi_{si}^l([0, r]) p_i(kr) \end{aligned} \quad (5)$$

Note that the matrix multiplications in (5) are the only online computations that are necessary to obtain the probabilistic reachable sets of the humans.

The discrete input l is chosen such that it includes the person state \hat{P} received from the tracker. In case of the *velocity model*, the chosen input l represents the cell \hat{z}_l which includes the heading angle β and the velocity $v([\beta, v]^T \in \hat{z}_l)$. As the velocity is a state variable in the *acceleration model*, \hat{z}_l is chosen in this case such that it includes the heading angle only. An exemplary probabilistic reachable set for the time interval case ($t \in [1s, 1.5s]$) using the *acceleration model* is shown in Fig. 6 (left), with initial state X_0 and velocity vector $v_\pi(0) = (\dot{x}(0), \dot{y}(0))^T$. To have a finite set of time intervals for the practical implementation, a maximal temporal prediction horizon T with $k \in \{0, 1, \dots, (\frac{T}{r} - 1)\}$ is introduced, where T is a natural multiple of r . Fig. 6 (right) shows all resulting reachable sets for $T = 3r$. How these reachable sets are used to perform the probabilistic mapping of the tracked persons is described in the following.

E. Low level Replanning

To enable safe and efficient navigation in crowded places, the static global path from the high level planning has to be adjusted by considering the dynamics of the persons. In this subsection a cost function applicable for a time minimizing path search is deduced. Focusing on a short-term adjustment of the global plan, the next waypoint of the latter is used as intermediate goalpoint c_{gp} for local replanning. First a

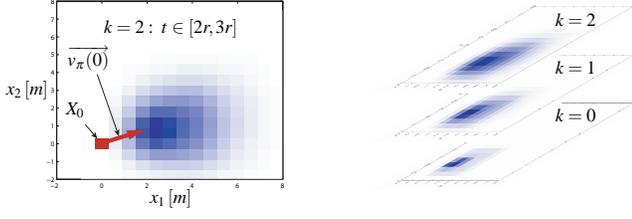


Fig. 6. Exemplary probabilistic reachable sets (acceleration model) for the following parameter set: $r = 0.5s$, $x(0) \in [-0.25m, 0.25m]$, $y(0) \in [-0.25m, 0.25m]$, $v(0) \in [1.6\frac{m}{s}, 2.4\frac{m}{s}]$ and $\beta(0) \in [0, \frac{\pi}{5}]$. Reachable set for $k = 2$ (left) and all reachable sets for $T = 3r$ (right).

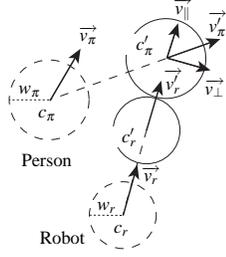


Fig. 7. Velocity obstacle.

potential field is computed, which provides the residual path costs W to c_{gp} . In parallel the reachable sets for each person π are computed and a velocity obstacle model is used to derive the delay costs D . A weighted sum of W and D leads to the combined costs C , which are used for greedy replanning.

1) *Computation of the residual path costs W* : Since the growing uncertainty for larger prediction intervals results in flatter distributions of the reachable sets, the prediction horizon T is chosen depending on the model dynamics to allow predictions within a reasonable time window. Given T and the desired velocity v_r of the robot, the local replanning is limited to the area A_{ll} , which is robot-centered and of quadratic size $(2Tv_r) \times (2Tv_r)$.

A wavefront algorithm as described in [25] is used to compute a local-minima-free potential field, providing the residual path costs $W = \{W_c | c \in A_{ll}\}$ where W_c is the path cost from cell c to c_{gp} . To provide a temporal metric the cost for moving from cell c_i to cell c_j is $\frac{d(c_i, c_j)}{v_r}$, where $d(c_i, c_j) = \|c_i, c_j\|_2 \in \mathbb{R}^+$ is the Euclidean distance of the respective cell centers. Furthermore $W_{c_{gp}} = 0$ is the unique global minimum.

2) *Computation of the delay costs D* : The probability vector $p_\pi([kr, (k+1)r])$ is computed for all k and each person π according to (5). The delay cost D_c^π in cell c occurring from person π , is given by the temporal delay resulting when the robot is forced to slow down or wait until the person moved out of the way. D is the set $\{D_c^\pi | c \in A_{ll}, \pi \in \Pi\}$ where Π is the set of perceived persons. To compute D_c^π the velocity obstacle approach shown in Fig. 7 is used. Taking the respective obstacle width w_π and robot width w_r into account and further assuming both to have circular shape, a potential collision occurs in case the robot moves from c_r to c'_r while the obstacle moves from c_π to c'_π , where $d(c'_\pi, c'_r) = 0.5(w_\pi + w_r)$. For this computation constant velocities are assumed, i.e. $v'_r = v_r$ and $v'_\pi = v_\pi$, while the heading $\beta'_\pi =$

$\angle(\overrightarrow{c_\pi c'_\pi})$ and $\beta'_r = \angle(\overrightarrow{c_r c'_r})$. Setting the velocity vector $\overrightarrow{v'_\pi}$ of person π into relation with the velocity vector $\overrightarrow{v'_r}$ of the robot, $\overrightarrow{v'_\pi}$ can be split into a parallel part $v_{\parallel} = |\overrightarrow{v'_\pi}|$ and an orthogonal part $v_{\perp} = |\overrightarrow{v'_\pi}|$, where

$$\overrightarrow{v'_\pi} = \overrightarrow{v_{\parallel}} + \overrightarrow{v_{\perp}}, \quad \text{with } \overrightarrow{v_{\parallel}} \parallel \overrightarrow{v'_r}, \quad \overrightarrow{v_{\perp}} \perp \overrightarrow{v'_r}. \quad (6)$$

Under the assumption that the people are avoiding the robot as well, it does not have to move backwards out of the way. So in case the robot meets person π , its possible speed v_r^π is bounded by $0 \leq v_r^\pi \leq \min(v_r, v_{\parallel}^\pi)$ for the duration of $\frac{1}{2v_{\perp}^\pi}(w_\pi + w_r)$ until the person moved at least $0.5(w_\pi + w_r)$ to the side. Such a deceleration leads – under assumption of constant velocities – to the delay cost

$$D_{c_r}^\pi = \frac{w_\pi + w_r}{2v_{\perp}^\pi} \left(1 - \frac{\max(\min(v_r, v_{\parallel}^\pi), 0)}{v_r} \right) \quad (7)$$

for meeting person π when being in cell c'_r . However, it has to be taken into account that $D_{c_r}^\pi$ applies only with probability

$$A_c^\pi = \frac{1}{\gamma} \mu_c^\pi(\tau_r), \quad \text{where} \quad (8)$$

$$\gamma = \left(\frac{\rho_c}{\rho_z} \right)^2 \sum \mu_v^\pi(\tau_r),$$

$$\tau_r = \left\{ [kr, (k+1)r] \mid \frac{d(c_r, c)}{v_r} \in [kr, (k+1)r] \right\}.$$

The probability μ_c^π ($\mu_c^\pi = \{0 | c \notin p^\pi\}$) is the entry of the probability vector p^π from (5) corresponding to cell c and τ_r is the time interval in which the robot would pass cell c . The right boundary of the conditional interval is excluded here to ensure an explicit assignment of cells to intervals. γ is a normalization factor to encounter cluttered environment. ρ_c and ρ_z are the respective cell resolutions of the occupancy grid and the reachable sets and v is the set of cells visible to person π . In that only the part of the reachable sets is encountered, that lies in free space and is in direct line of sight to person π .

3) *Computation of the costs C* : For the path replanning the path costs W and the delay costs D have to be both considered. The combined cost

$$C_c = \alpha W_c + (1 - \alpha) \sum_{\pi} A_c^\pi D_c^\pi \quad (9)$$

of traversing cell c results from a weighted sum of its path cost W_c and its overall delay cost D_c where $\alpha \in]0, 1[$ is a weighting factor providing different replanning strategies. A small α corresponds to a low risk strategy, resulting in a path primarily avoiding people while a large α corresponds to a strategy preferring short path length and accepting higher collision risks. The lower interval boundary has been excluded, since $\alpha = 0$ leads to $C_c = 0$ in all cells which do not intersect with the reachable sets of dynamic objects and consequently results in a violation of the constraint for $C = \{C_c | c \in A_{ll}\}$ to have a single global minimum.

To adjust the global path, a simple greedy search on C is used for replanning. Since C can have local minima within risk regions, hill-climbing is enabled by limiting the search from cell c_i to the set of neighboring cells c_n for which $W_{c_n} \leq W_{c_i}$. In the next section, some experimental results for

TABLE I
MODEL PARAMETERS

<i>Robot:</i>	$v_r = 2 \frac{m}{s}$
<i>Resolutions:</i>	$\rho_c = 0.15m$ $\rho_z = 0.6m$ $\Delta t = 0.5s, T = 3s$
<i>Dynamic model:</i>	$v \in [0 \frac{m}{s}, 2.4 \frac{m}{s}]$ $\Delta v = 0.8 \frac{m}{s}$ (3 sectors) $\Delta \beta = \frac{\pi}{5}$ (10 sectors) $a \in [-1.0 \frac{m}{s^2}, 1.0 \frac{m}{s^2}]$ $\Delta a = \frac{2}{3} \frac{m}{s^2}$ (3 sectors)
<i>only acceleration model</i>	$v_\pi = 1 \frac{m}{s}$
<i>Simulated Persons (ground truth):</i>	
<i>Replanning:</i>	
<i>short distance path (cyan)</i>	$\alpha = 1$
<i>fast path (blue)</i>	$\alpha = 10^{-5}$
<i>low risk path (green)</i>	$\alpha = 10^{-18}$

TABLE II
REPLANNING DURATION (AVERAGED OVER 100 RUNS)

No. of Persons	Velocity model		Acceleration model	
	$\mu \pm \sigma$	max	$\mu \pm \sigma$	max
1	45ms \pm 4ms	50ms	100ms \pm 4ms	109ms
2	85ms \pm 4ms	108ms	198ms \pm 4ms	214ms
3	117ms \pm 5ms	127ms	308ms \pm 6ms	322ms

various situations are given, showing risk maps and resulting path for different values of α .

IV. EXPERIMENTAL RESULTS

The described algorithm has been fully implemented in C++ as part of the ACE project [1]. Fig. 8 shows results derived online with the ACE robot. Given that real data suffers from various sources of uncertainty, the remaining results (Fig. 9 and Fig. 10) have been derived using the Stage simulator of the Player Project (see <http://playerstage.sourceforge.net>). Thus, ground truth for the person states P could be used, giving the results an enforced stand-alone character. To achieve comparability of the scenes, a constant v_r was used only for replanning, meaning the robot did not move during the shown simulations. Table I contains the parameters used for the simulations.

Fig. 9 shows the risk regions for the *velocity* and *acceleration model* for a scene with three persons in free space. Replanning with constant v_r and constant Δt leads to an uniform spatial interval propagation around the robot, observable in the slight ring structure of the risk regions. Fig. 9 shows furthermore the resulting paths for replanning with three values of α , revealing how the value of α is influencing the magnitude of deviation from the shortest path. The paths for $\alpha = 1$ are disregarding the delay costs D completely, resulting in paths with shortest length. The triangular shape results from the cell-wise replanning. Replanning with $\alpha = 10^{-18}$ indicates the opposite case where the path costs W are almost neglected and the delay costs D are given a strong weight, resulting in low risk paths trying to avoid the risk regions as far as possible. The paths with $\alpha = 10^{-5}$ are searching for a trade-off between goal-orientation and risk minimization. The small order of magnitude of α arises from the broad and therefore flat risk distributions and furthermore from the γ -normalization, which considers the mapping of one z cell to here 16 corresponding c cells.

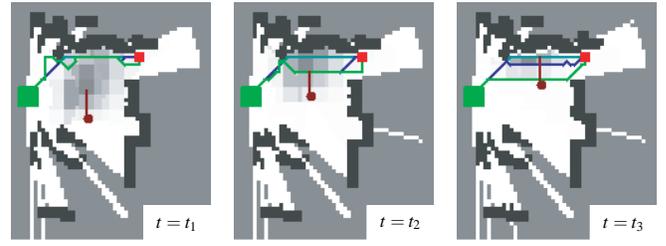


Fig. 8. Scene with one person for three time steps in cluttered environment (C-Space), derived online with the ACE robot (*acceleration model*).

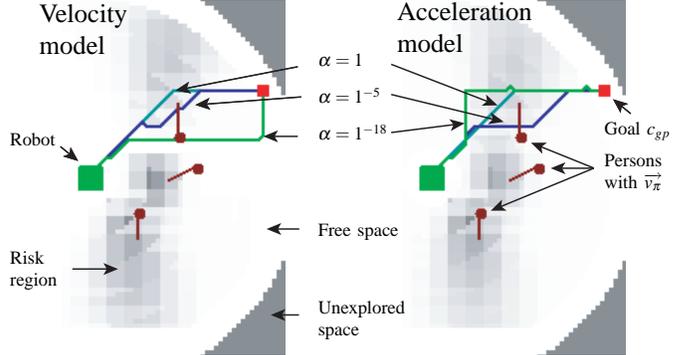


Fig. 9. Scene with three persons in free space: comparison of *velocity* and *acceleration model* for different values of α .

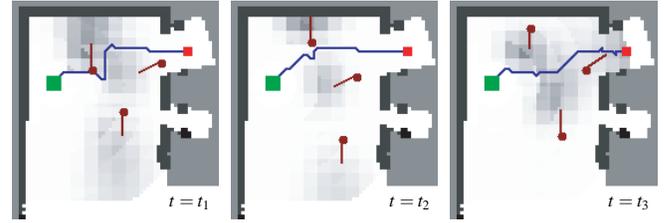


Fig. 10. Scene with three persons in cluttered environment for three time steps (*acceleration model*, $\alpha = 10^{-5}$).

Comparing in Fig. 9 the *velocity model* (left) with the *acceleration model* (right), shows broader and flatter risk regions for the latter case. This can be traced back to the extended model dynamics, which consider also the possibility of a turn around in case of a deceleration. This becomes very clear when considering the low risk paths for $\alpha = 10^{-18}$. For the *velocity model* the risk regions of the upper and the middle person do not intersect, leading to a feasible low risk path passing between them. The risk regions derived with the *acceleration model* do intersect. Therefore the low risk path deviates in the beginning to the top, until the hill-climbing behavior forces it to pass in front of the upper person.

Fig. 10 points out the effect of the γ -normalization. Here for the upper person heading towards the wall at the top ($t = t_1 \wedge t = t_2$), only the parts of the reachable sets lying below the wall are considered. This leads to higher risk potentials in front of the wall compared to the risk regions corresponding the other two persons.

The softness attribute of the approach is especially perceivable at $t = t_3$, when considering the person heading to the right through the upper door. In this situation, hard obstacle avoidance or space-time volumes might block the

door resulting in no feasible path at all, while the presented approach is still capable to find a valid path.

All results were derived on a AMD Athlon 64 X2 Dual Core 3800+ with 2GHz and 2GB RAM. The average and maximal replanning durations for the scenario of Fig. 10 are shown in Table II. The complexity to compute W scales linear with the size $(2Tv_r)^2$ of A_{ll} . The complexity to compute D scales linear with the number of persons Π , where the complexity to compute A_c^π scales quadratic with the number of acceleration sectors u ($u = 1$ for the *velocity model*). Thus the complexity to compute the costs C results in $O((2Tv_r)^2 + \Pi u^2)$.

V. CONCLUSION

An approach for a risk mapping of moving persons is presented which leaves the complete space free of static objects for path planning. The dynamics of people are modeled offline with Markov Chains, which are used online to predict reachable sets of tracked persons and provide an estimate of their potential future locations. These estimates are then combined with a velocity obstacle approach to determine the delay costs D , resulting in case of a potential co-occurrence of robot and respective person. A weighted sum of delay costs D and path costs W serves to adjust a global static plan by considering the environmental dynamics. The weighting factor α provides the soft trade-off between detour and time delay for replanning.

The results show the capability of the approach to find a valid path through potentially obstructed space in densely populated environments. Furthermore the suitability for on-line applications could be shown. Although the approach is here applied to people, it can be easily extended to other dynamic objects.

Future work may concentrate on extended MC models considering also interactions of dynamic objects with the environment. Further interesting developments especially with focus on fast replanning are to combine the delay costs D with anytime path planning solutions such as AD*. Another important concern is that object occlusion often causes the data association of the tracker to fail, resulting in inaccurate state estimations of the persons. More reliability in this respect would considerably increase the robustness of the presented approach.

VI. ACKNOWLEDGMENTS

This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys*, see also www.cotesys.org.

REFERENCES

- [1] G. Lidoris, K. Klasing, A. Bauer, T. Xu, K. Kühnlenz, D. Wollherr, and M. Buss, "The autonomous city explorer project: Aims and system overview," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2007.
- [2] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 878–885, IEEE Computer Society, 2005.
- [3] K. Kühnlenz, M. Bachmayer, and M. Buss, "A multi-focal high-performance vision system," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 150–155, 2006.
- [4] S. Bahadori, L. Iocchi, G. R. Leone, D. Nardi, and L. Scozzafava, "Real-time people localization and tracking through fixed stereo vision," in *Proc. of the Int. Conf. on Innovations in Applied Artificial Intelligence*, pp. 44–54, 2005.
- [5] M. Montemerlo, W. Whittaker, and S. Thrun, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2002.
- [6] G. Lidoris, D. Wollherr, and M. Buss, "Bayesian state estimation and behavior selection for autonomous robotic exploration in dynamic environments," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2008.
- [7] R. Philippsen and R. Siegwart, "Smooth and efficient obstacle avoidance for a tour guide robot," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003.
- [8] M. Zucker, J. Kuffner, and M. Branicky, "Multipartite RRTs for rapid replanning in dynamic environments," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2007.
- [9] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime dynamic A*: An anytime, replanning algorithm," in *Proc. of the Int. Conf. on Automated Planning and Scheduling*, 2005.
- [10] J. van den Berg, D. Ferguson, and J. Kuffner, "Anytime path planning and replanning in dynamic environments," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 2366–2371, 2006.
- [11] K. Klasing, D. Wollherr, and M. Buss, "Cell-based probabilistic roadmaps (CPRM) for efficient path planning in large environments," in *Proc. of the 2007 Int. Conf. on Advanced Robotics*, 2007.
- [12] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, pp. 760–772, 1998.
- [13] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2007.
- [14] F. Lamarche and S. Donikian, "Crowd of virtual humans: a new approach for real time navigation in complex and structured environments," in *Computer Graphics Forum*, vol. 23, pp. 509–518, 2004.
- [15] M. Lau and J. J. Kuffner, "Behavior planning for character animation," in *2005 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pp. 271–280, 2005.
- [16] D. Vasquez, F. Large, T. Fraichard, and C. Laugier, "High-speed autonomous navigation with motion prediction for unknown moving obstacles," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 82–87, 2004.
- [17] R. Philippsen, B. Jensen, and R. Siegwart, "Toward online probabilistic path replanning in dynamic environments," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2006.
- [18] R. Philippsen and R. Siegwart, "An interpolated dynamic navigation function," in *Proc. of the 2005 IEEE Int. Conf. on Robotics and Automation*, 2005.
- [19] M. Bennis, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The Int. Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, 2005.
- [20] M. Althoff, O. Stursberg, and M. Buss, "Safety assessment of autonomous cars using verification techniques," in *Proc. of the American Control Conference (ACC)*, 2007.
- [21] M. Althoff, O. Stursberg, and M. Buss, "Online verification of cognitive car decisions," in *Proc. of the 2007 IEEE Intelligent Vehicles Symposium*, 2007.
- [22] M. Lopez-Sanchez, G. Sukhatme, and G. A. Bekey, "Mapping an outdoor environment for path planning," tech. rep., Research Report from the University of Southern California (USA), 1998.
- [23] T. Henzinger, *The theory of hybrid automata*, vol. 170 of *NATO ASI Series F: Computer and Systems Sciences*, pp. 265–292. Springer, 2000.
- [24] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of linear systems with uncertain parameters and inputs," in *Proc. of the Conf. on Decision and Control*, 2007.
- [25] J. Barraquand, B. Langlois, and J. Latombe, "Numerical potential field techniques for robot path planning," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, pp. 224–241, 1992.