Qualitative Constraints for Human-aware Robot Navigation using Velocity Costmaps*

Christian Dondrup[1] Marc Hanheide[2]

***Abstract*— In this work, we propose the combination of a state-of-the-art sampling-based local planner with so-called *Velocity Costmaps* to achieve human-aware robot navigation. Instead of introducing humans as "special obstacles" into the representation of the environment, we restrict the sample space of a "Dynamic Window Approach" local planner to only allow trajectories based on a qualitative description of the future unfolding of the encounter. To achieve this, we use a Bayesian temporal model based on a Qualitative Trajectory Calculus to represent the mutual navigation intent of human and robot, and translate these descriptors into sample space constraints for trajectory generation. We show how to learn these models from demonstration and evaluate our approach against standard Gaussian cost models in simulation and in real-world using a non-holonomic mobile robot. Our experiments show that our approach exceeds the performance and safety of the Gaussian models in pass-by and path crossing situations.**

I. INTRODUCTION

With mobile robots advancing into our daily lives not only in public and work places but also in private homes, the ability to navigate and manoeuvre safely around humans becomes ever more important [1]. Mobile robots currently used are able to navigate safely throughout their environment, avoiding not only static but also dynamic obstacles quite reliably. But due to humans, for example, requiring a greater distance during circumvention to feel safe and comfortable [2], treating people as dynamic obstacles and merely avoid them is not sufficient. Human-Robot Spatial Interaction (HRSI), as the study of joint movement of robots and humans through space and the social signals governing these interactions, is therefore concerned with the investigation of models of ways humans and robots manage their motions in vicinity to each other. Typical encounters occurring in the daily life of a mobile robot might, for example, be so-called *pass-by* situations where human and robot aim to pass through a corridor in opposite direction, trying to circumvent each other given spatial constraints. In order to resolve these kinds of situations and pass through the corridor, the human and the robot need to be aware of their mutual goals and have to have a way of negotiating, e.g. prompting [3], who goes first or who goes to which side (Fig. 1 shows and example of such an interaction). Our work aims to equip a mobile robot with the ability to capture the trace of movements of not only the human but also itself to better be able to represent and reason about mutual intent and to act accordingly.

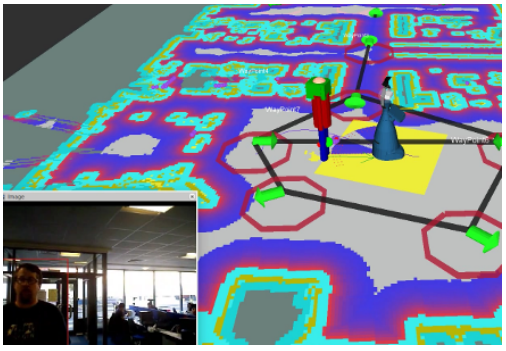


Fig. 1. Using Velocity Costmaps based on QTC descriptors in an office environment with our SCITOS G5 mobile robot.

Over the last 20 years, the robotics community started to take the dynamic aspects of "human obstacles" into account, with [4] being one of the earliest examples. Currently a large body of research is dedicated to answer the more fundamental questions of HRSI to produce navigation approaches which plan to explicitly move on more "socially acceptable and legible paths" [5]. The term "legible" here refers to the communicative – or interactive – aspects of movements which previously has widely been ignored in robotics research. This legibility is especially important because humans do not only take their own actions into account when planning their path but also the actions of other mobile agents in their close vicinity as shown by Ducourant *et al.* [6] who investigated human-human spatial interaction. We therefore build on our previous work [7] introducing a Qualitative Trajectory Calculus (QTC) [8] for the representation of HRSI. This calculus allows to encode the actions of human and robot in the same framework and hence represents their actions in relation to each other.

State-of-the-art navigation approaches currently mostly rely on a combination of global and local planning, e.g. [9], [10][1], to achieve robust navigation in the face of static and dynamic obstacles. For dynamic obstacle avoidance, a cost model is produced in metric space and then translated to the velocity space to allow for dynamic sampling of future trajectories. In contrast to this, our QTC based representation encodes the interaction of human and robot directly in the velocity space. Therefore, we introduce so-called *Velocity*

*The research leading to these results has received funding from the European Community's FP7 Programme under grant agreement No. 600623, STRANDS, and the European Unions H2020 programme under grant agreement No. 688147, MuMMER http://mummer-project.eu/.

[1]Christian Dondrup, School of Mathematical and Computer Sciences, Heriot-Watt University, EH14 4AS, Edinburgh, UK C.Dondrup@hw.ac.uk

[2]Marc Hanheide, School of Computer Science, University of Lincoln, LN6 7TS, Lincoln, UK mhanheide@lincoln.ac.uk

[1]As implemented in the popular and widely used Robot Operating System (ROS) navigation stack: http://wiki.ros.org/navigation

Costmaps used in conjunction with a Dynamic Window Approach (DWA) local planner [9] to achieve robust and informed human-aware navigation. Given this qualitative framework, we are able to learn QTC based rules from Demonstration. Hence, we are creating models of specific types of interactions using human judgement on how a mobile robot should react to the interaction partner.

The main contributions of this work, therefore, are *Velocity Costmaps* based on a qualitative state description to restrict the sample space of a DWA local planner to generate trajectories that, on the one hand, are safe and also perceived as safe by the human interaction partner and, on the other hand, are still able to minimise time and distance travelled towards the goal. To achieve this trade-off, we use human judgement on the execution of these interactions by learning from demonstration. We are evaluating our human-aware navigation framework in simulation and a real-world experiment using a non-holonomic robot, showing how to incorporate knowledge about HRSI into a concise model for trajectory sampling in velocity space. The presented software is freely available as open source or precompiled Debian packages[2].

II. Related Work

Qualitative spatial representations like QTC are used on a large scale in many different research areas and fields [11] but are novel to the field of human-aware navigation [12]. In our case, QTC states are used to describe interactions between a human and a robot in the spatial domain, *i.e.*, 2D navigation [7]. With the presented approach, we employ these representations for path planning in the vicinity of humans. Path planning for mobile robots in general aims at finding a safe and short path which, in the majority of cases, is done by some form of Dijkstra or A* algorithm. HRSI on the other hand, does not aim to find the shortest or most energy efficient path like the aforementioned algorithms but tries to adhere to numerous social norms and conventions, like the concept of personal and social space defined as proxemics [13]. Thereby, they arguably make navigation in human-populated environments safer and more efficient. There are many ways of solving this problem like Social Force Models, e.g. [14], or Trajectory Learning, e.g. [15] but the majority of human-aware path planners relies on specific cost functions or potential fields, mainly circular or elliptical Gaussians, e.g. [5], [16], [17]. These approaches all rely on constraints or observed interactions and represent previous encounters via definitions to create or tune cost functions and potential fields. These representations or cost functions are then applied to local path planning using computationally cheap sampling based approaches like the popular and widely used DWA [9] local planner on a so-called local costmap and can easily be combined with global planners targeted at HRSI like [18]. This local costmap approach however restricts the behaviour generation to be purely reactive to humans appearing in the immediate vicinity of the robot and therefore deprives it of most of its legibility. Our approach on the other hand uses the same planning framework but encodes high-level knowledge of the unfolding of the interaction to restrict the sampling space of the planner without needing the human as a special obstacle.

[2]See `http://www.dondrup.net` for a concise overview or `http://lncn.eu/strands` for installation instructions.

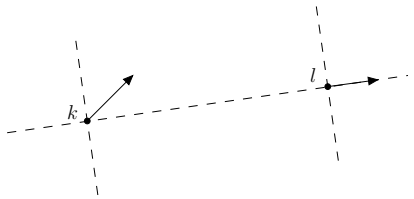


Fig. 2. The QTC_C double cross. The respective QTC_B and QTC_C relations for k and l are $(-+)$ and $(-+-0)$.

Learning from Demonstration (LfD) is a popular principle in robotics, creating policies from example state to action mappings [19]. In human-aware navigation there are only very few examples that make use of this approach. In [20] the authors use trajectories recorded while being guided through an unknown environment to achieve reliable navigation exploiting the human's knowledge about the environment. In [21] a naïve participant is tele-operating the robot to record the preferred trajectories in path-crossing situations. Both of these approaches make use of the knowledge and experience of the human demonstrator where the latter also uses a QTC based representation but only to classify the recorded trajectories offline and not for online behaviour generation.

III. The Qualitative Trajectory Calculus

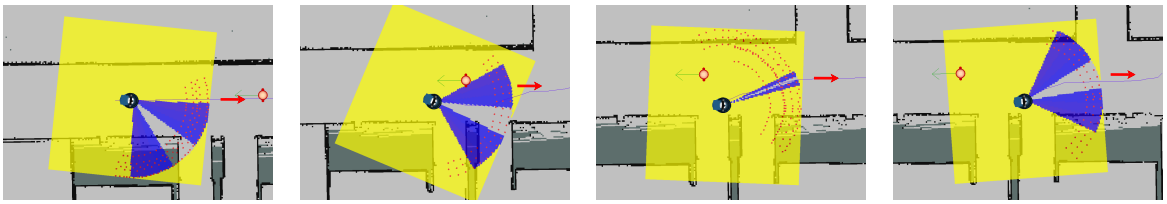
We model HRSI as a probabilistic sequential model of qualitative states, describing the relative motion of two agents in a 2D world. In particular, to model the states, we use the Qualitative Trajectory Calculus (QTC) which belongs to the broad research area of qualitative spatial representation and reasoning [11], from which it inherits some of its properties and tools. The calculus was originally developed by Van de Weghe in 2004 to represent and reason about moving objects in a qualitative framework [8] and has since been used in a broad area of applications, e.g. [7]. There are several versions of QTC of which the most important variants for our work are explained in the following.

A. QTC Basic and Double Cross

QTC in general represents the relative movement of two points k and l over the interval $[t_n, t_{n+1}]$, i.e. $k_{t_{n+1}}$ compared to l_{t_n} and vice-versa. The simplest version, called QTC Basic (QTC_B), represents the 1D relative motion of these two points (see Fig. 2). It uses a 2-tuple of qualitative relations $(q_1\ q_2)$, where each element can assume any of the values $\{-, 0, +\}$ as follows:

q_1) movement of k with respect to l
- $-$: k is moving towards l
- 0 : k is stable with respect to l
- $+$: k is moving away from l

q_2) movement of l with respect to k: as above, but swapping k and l



(a) $(+\ 0\ -\ 0) \rightarrow (-+)$ (b) $(+\ 0\ -\ +) \rightarrow (-+)$ (c) $(+\ 0\ 0\ +) \rightarrow (0\ +)$ (d) $(+\ 0\ +\ +) \rightarrow (++)$

Fig. 3. Example of a pass-by interaction. Blue figure: robot, red figure: human. The partial circles (with radius $\max(\rho)$) inside the yellow square represent a Cartesian representation of the Polar space used for the Velocity Costmap (see Fig. 4). Blue: low cost areas $\{5, 10, 15\}$ to increase avoidance manoeuvre (see Equ. 1), yellow: maximum costs of 100, free space: 0 costs, red dots: generated samples $\tau_j \in \boldsymbol{\tau}$. Captions represent the mapping $O_j \rightarrow S_i$ of observed human state to learned robot state. The red arrow points towards the robot's goal which is not visible in the images.

Hence, QTC$_B$ models the states of attraction, repel, and stationary and is defined as $\{(q_1\ q_2) : q_j \in \{-, 0, +\}\}$.

The other version of the calculus used in our model, called QTC Double-Cross (QTC$_C$) for 2D movement, extends the previous one to include also the side the two points move to, i.e. left, right, or along the connecting line $\overrightarrow{k\ l}$, $\overrightarrow{l\ k}$ (see Fig. 2)[3]. In addition to the 2-tuple $(q_1\ q_2)$ of QTC$_B$, the relations $(q_3\ q_4)$ are considered, where each element can also assume any of the values $\{-, 0, +\}$ as follows:

q_3) movement of k with respect to $\overrightarrow{k\ l}$
- $-$: k is moving to the left side of $\overrightarrow{k\ l}$
- 0 : k is moving along $\overrightarrow{k\ l}$
- $+$: k is moving to the right side of $\overrightarrow{k\ l}$

q_4) movement of l with respect to $\overrightarrow{l\ k}$: as above, but swapping k and l

Hence, QTC$_C$ is defined as $\{(q_1\ q_2\ q_3\ q_4) : q_j \in \{-, 0, +\}\}$.

B. Combining QTC$_B$ and QTC$_C$

As proposed in previous work [7], we combine QTC$_B$ and QTC$_C$ into the joint model QTC$_{BC}$ based on the Euclidean distance $d(k, l)$ between the two agents. This results in $\{(q_1\ q_2\ q_3\ q_4) : q_1, q_2 \in \{-, 0, +\}, q_3, q_4 \in \{-, 0, +, \emptyset\}\}$ where $q_3, q_4 = \emptyset$ if $d(k, l) > d_s$ where d_s is a predefined distance threshold. The reasoning behind this being that when k and l are far apart, we are only interested in knowing if either k or l are respectively approaching the other or not for noise reduction and to highlight the "essence" of the interaction in close proximity. In previous work [7], we showed that for pass-by scenarios in HRSI a distance threshold of $d_s \geq 1.8m$ is sufficient to reliably classify passing on the left or right which means that this threshold can be freely chosen or learned as long as $d_s \geq 1.8m$ holds.

IV. System Architecture

The basis for the system is a human tracker and QTC state generator which we introduced in previous work [22] which produces human tracks at 30Hz and QTC states at 3Hz due to smoothing using $\sim 30\%$ of a single core of an Intel i7 processor. The generated QTC states are used to find the next best action for the robot given the current observation of the human and learned behaviour model. The desired robot state is then passed to the Velocity Costmap server which creates an occupancy map representing the desired state as costs in velocity space (see Fig. 3 and 5) which is fed to the local planner.

A. QTC based HRSI Activity Modelling

The models to find the next best action for the robot use a conglomerate of different QTC states. We produce states in QTC$_{BC}$ for *human* (h) and *robot* (r) to also encode the distance between the two and QTC$_C$ states for the human and the *robot's goal* (g). This allows us to not only model the interaction between human and robot but also the intention of the robot by including the robot's goal. The resulting QTC states for each observation, therefore, consist of the QTC$_{BC}$ 4-tuple $(q_1^{hr}\ q_2^{hr}\ q_3^{hr}\ q_4^{hr})$ representing the state of human and robot and the QTC$_C$ 4-tuple $(q_1^{hg}\ q_2^{hg}\ q_3^{hg}\ q_4^{hg})$ representing the relative movement of human and the robot's goal. The symbols $q_1^\star$ and $q_3^\star$ represent the movement of the human and $q_2^\star$ and $q_4^\star$ represent the robot or the robot's goal. Since the robot's goal does not move during the interaction, we are disregarding $(q_2^{hg}\ q_4^{hg})$ in the following. Using the 4 symbols describing the human movement, we create the current observation $O_j = (q_1^{hg}\ q_3^{hg}\ q_1^{hr}\ q_3^{hr})$ and use the remaining two symbols to describe the state of the robot $S_i = (q_2^{hr}\ q_4^{hr})$. The mapping of the current observation of the human to the robot state can, therefore, be expressed as $O_j \rightarrow S_i$. Hence, the sum of all observations results in the two sets of states $\boldsymbol{\Omega} = \{(q_1^{hg}, q_3^{hg}, q_1^{hr}, q_3^{hr}) : q_1^{hg}, q_3^{hg}, q_1^{hr} \in \{-, 0, +\}, q_3^{hr} \in \{-, 0, +, \emptyset\}\}$ and $\boldsymbol{\Sigma} = \{(q_2^{hr}, q_4^{hr}) : q_2^{hr} \in \{-, 0, +\}, q_4^{hr} \in \{-, 0, +, \emptyset\}\}$ with $|\boldsymbol{\Omega}| = 108$, $|\boldsymbol{\Sigma}| = 12$ and $O_j \in \boldsymbol{\Omega}$, $S_i \in \boldsymbol{\Sigma}$.

To predict the most appropriate robot state S_i using our mapping, we create the conditional probability table $P(S_i|O_j)$ by simply counting occurrences of $O_j \rightarrow S_i$ with $O_j \in \boldsymbol{\Omega}$ and $S_i \in \boldsymbol{\Sigma}$. The resulting state space for all possible combinations is therefore $\boldsymbol{\Sigma} \times \boldsymbol{\Omega}$ of which only a fraction is observed for each interaction.

B. Velocity Costmaps

In this work, we use the DWA local planner [9][4] which we consider state-of-the-art because it is part of the default Robot Operating System (ROS) navigation stack, which is widely used and very popular with many research and

[3]The actual variants of QTC described here are QTC$_{B11}$ and QTC$_{C21}$ to which we will from here on refer to as QTC$_B$ and QTC$_C$ for simplicity.

[4]`http://wiki.ros.org/dwa_local_planner`

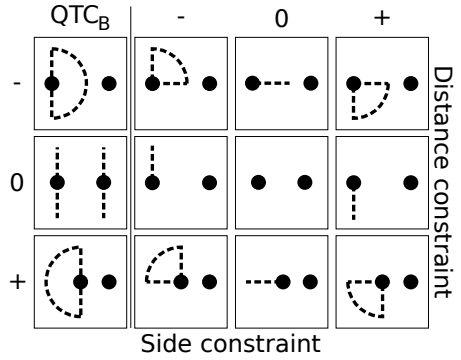


Fig. 4. The velocity costmap prototypes. The area enclosed by the partial circle represents the low cost area ξ, everything outside is assigned the highest cost value. The black dot on the right represents a human that can have any possible QTC state (except for QTC$_B$ 0). These zones are directly inspired by the original definition of QTC [8].

industrial projects all around the world. This planner samples trajectories in velocity space to avoid obstacles which, for a non-holonomic robot, is equivalent to the Polar coordinate space (ρ, θ) where ρ represents the linear and θ the angular velocity. Hence, the set of all samples is defined as $\boldsymbol{\tau} = \{(\rho_i, \theta_i) : \rho_i \in \boldsymbol{\rho}, \theta_i \in \boldsymbol{\theta}\}$ with $\boldsymbol{\rho}$ and $\boldsymbol{\theta}$ being the set of all possible angular and linear velocities given the current sample granularity of the planner. Each trajectory $\tau_j \in \boldsymbol{\tau}$ is assigned a cost value $\gamma(\tau_j)$ based on the sum of several independent, weighted critique functions $\gamma(\tau_j) = \sum_{i=0}^{N} \omega_i c_i(\rho_j, \theta_j)$ where $c_i \in \boldsymbol{\mathcal{C}}$ and $\omega_i \in \boldsymbol{\omega}$ denote the critique function and its associated weight and $N = |\boldsymbol{\mathcal{C}}|$. Using the dynamic window, the sample space $\boldsymbol{\tau}$ is restricted to velocities that can be reached in a given time window based on the acceleration limits of the robot. The trajectory $\arg\min_{\tau_j \in \boldsymbol{\tau}} \gamma(\tau_j)$ is then executed until the next sampling step.

In this work, we propose the addition of QTC based Velocity Costmaps for human-aware navigation into the set of critique functions $\boldsymbol{\mathcal{C}}$. Instead of using a Cartesian cost representation, we operate directly in QTC's velocity space (see Fig. 4) by assigning costs to samples τ_j that do not fit the predicted state the robot should assume during the interaction, e.g. only allowing the robot to pass a human on the right side. Since this critique function is included into the set of critique functions $\boldsymbol{\mathcal{C}}$, it allows to determine the human awareness of the local planner by adjusting the weights $\boldsymbol{\omega}$ accordingly. By assigning a higher weight to the obstacle avoidance, for example, we ensure the collision free navigation of the robot while avoiding humans.

Fig. 3 and 5 show exemplary encounters using the Velocity Costmaps. Note, these images show a Cartesian representation of the underlying Polar velocity space. Hence, the circle with radius equivalent to the maximum translational speed of the robot $\max(\rho)$ inside this rectangular map is used in the sampling process. Therefore, the size of the map and red dots representing the samples $\boldsymbol{\tau}$ are completely independent of the underlying metric map as they represent $(\boldsymbol{\rho}, \boldsymbol{\theta})$.

Using the Velocity Costmaps critique function c_{vc}, given the current position of the human (ρ_h, θ_h) relative to the robot, we compute the angle $\lambda = \theta_h + \delta$ where δ depends on the desired QTC state of the robot. If τ_j lies within the allowed area of $\xi_\alpha = \{\theta_j \in \mathbb{R} : \lambda - \alpha \leq \theta_j \leq \lambda + \alpha\}$ and $\xi_\varrho = \{\rho_j \in \mathbb{R} : \varrho_{\min} \leq \rho_j \leq \varrho_{\max}\}$ where α an ϱ, like δ, also depend on the desired QTC state (see Tab. I)[5], we compute for each trajectory sample $\tau_j(\rho_j, \theta_j)$

$$c_{vc}(\tau_j) = \boldsymbol{a}_{||\theta_j| - |\lambda||\cdot|\boldsymbol{a}|};\ \boldsymbol{a} = \{a_i : a_i \in \mathbb{N}\} \qquad (1)$$

with $\boldsymbol{a}$ being a strictly increasing set of low costs. If $\tau_j \notin \xi$ then $c_{vc}(\tau_j)$ is set to the maximum costs of 100. Looking at Fig. 3(a) as an example, given the desired QTC state of $(-+)$ approaching and moving to the right, we compute $\lambda = \theta_h + \frac{\pi}{4}$ and get an allowed sample space of $\xi_\alpha = [\lambda - \frac{\pi}{4}, \lambda + \frac{\pi}{4}]$. Assuming that the human is directly infront of the robot $\theta_h = 0.0$, we get $\lambda = \frac{\pi}{4}$ and $\xi_\alpha = [0, \frac{\pi}{2}]$ as it is shown in the top right corner of Fig. 4. We then use $\boldsymbol{a} = \{0, 5, 10, 15\}$ as low cost areas (see blue areas in Fig. 3 and 5) to increase the avoidance manoeuvre. The relatively low cost values of $\boldsymbol{a}$ allow to give a small bias to the center of the region while not drastically restricting the search space for collision free trajectories. This results in a relatively hard transition from the highest low cost value to lethal costs but every sample outside the designated area should never be considered for execution. The resulting costs are then weighted and summed with the remainder of the critique functions. Given this representation, it is also possible to restrict the minimum and maximum speed of the robot in addition to the angular speed but this is currently only used for the QTC$_B$ state $(0\ 0)$ to allow the robot and human to travel in the same direction with equal velocity. If the human is not observed any more, an empty map $c_{vc}(\tau_j) = 0$ is assumed, allowing the DWA planner to use the whole sample space. Velocity Costmap creation itself takes $\sim 0.001s$ and the sampling process is a simple look-up in a 2D array.

TABLE I

THE δ AND α VALUES TO COMPUTE THE VELOCITY COSTMAPS

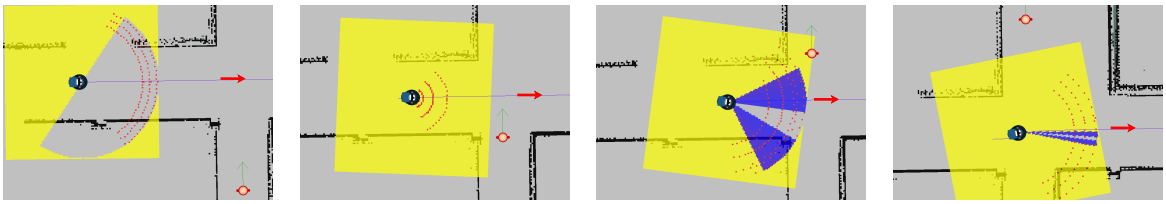
	QTC$_B$	$-$	0	$+$	
$-$	0	$-\pi/4$	0	$\pi/4$	δ
	$\pi/2$	$\pi/4$	$\pi/32$	$\pi/4$	α
0	$\pm\pi/2$	$-\pi/2$	0	$\pi/2$	δ
	$\pi/32$	$\pi/32$	0	$\pi/32$	α
$+$	π	$-3\pi/4$	π	$3\pi/4$	δ
	$\pi/2$	$\pi/4$	$\pi/32$	$\pi/4$	α

V. EXPERIMENT AND EVALUATION

To evaluated the functionality and soundness of our proposed velocity costmaps using QTC$_{BC}$, we conducted an experiment in simulation and a proof of concept experiment using a SCITOS G5 mobile robot.

We constructed a simulated office environment of $\sim 5,000m^2$ resembling one of our university buildings, using its main corridor for our experiment. We created a topological edge along a 12m long and 2.6m wide straight

[5] ϱ is not shown as it is only used in the $(0\ 0)$ case where it is set to $\varrho = \rho_h \pm \Delta$ with $\Delta = 0.05^m/_s$. Otherwise it ranges from 0 to the maximum translational velocity of the robot.



(a) $(- - -) \rightarrow (-)$ (b) $(- - -+) \rightarrow (0\ 0)$ (c) $(+ - ++) \rightarrow (-+)$ (d) $(+ - +\ 0) \rightarrow (0\ +)$

Fig. 5. Example of a path crossing sequence where "$\rightarrow$" represents the mapping from observed state to generated constraint. The transition from QTC_B 5(a) to QTC_C 5(b) causes a state change in the robot even though the human's state is unchanged. See Fig. 3 for an explanation of the symbols.

stretch and another 15m long edge passing a 4-way crossing (see Fig. 3 and 5). No obstacles but walls and the physical model of the simulated human are present in these parts of the environment. For the real world environment we used a 8m x 8m area of our office that was cleared of all obstacles but the human interaction partner, and defined a 5.5m long topological edge passing through the centre of the free area (see Fig. 1).

Initially, the robot was remote controlled by the experimenter while interacting with the real or simulated human. The recorded QTC states were then transformed into the joint probability table $P(S_i|O_j)$ by counting the occurrences of $O_j \rightarrow S_i$. The behaviours shown to the robot during these learning sessions were to avoid people to the right in pass-by encounters (see Fig. 3) and to stop and wait in path crossing situations (see Fig. 5). In the subsequent evaluation, the robot showed 4 different behaviours based on the chosen avoidance cost model: *i)* the vanilla DWA planner, *ii)* a Gaussian Cost model on the local map (G-Local), *iii)* a Gaussian Cost model on the global map (G-Global), and *iv)* our proposed Velocity Costmap QTC_{BC} approach (Vel-Maps). The weights of the used critique functions were: *Velocity Costmaps:* 30, *Oscilation:* 1, *Goal Align:* 10, *Path Align:* 10, *Goal Distance:* 24, *Path Distance:* 10, *Obstacles:* 0.01 (only lethal obstacles) and 30 when using G-Local which proved to work the best using trial and error.[6] The QTC_{BC} distance threshold was set to $d_s = 4.0m$.[7] All the parameters were the same in simulation and on the real robot. The main difference between the experiments was the full observability of the human in simulation compared to the perception pipeline described in [22] which only tracks the human in an area of up to 7m and 180 degrees in front of the robot.

In both experiments the robot was reset to its original starting position and traversed the edge in the same direction towards the same goal using one of four planner variants. The simulated and real human also started from always the same position and moved towards the goal. The simulated human received a constant velocity command of $\rho = 0.55^{m}/_{s}, \theta = 0.0^{rad}/_{s}$ which corresponds to the robots maximum linear velocity. For the pass-by scenario both robot and human moved on a straight line towards each other, whereas during the path crossing the human's position was offset by $90°$ to create perpendicular trajectories (see Fig. 6). In both cases, if the robot did not initiate an avoidance behaviour, robot and human would collide half way through traversing the edge. The same conditions were recreated in the real world experiment. Participants walked on a straight line towards a marker on the other side of the room. The starting positions for pass-by were slightly offset to the right to account for a later detection of the human but would still lead to a collision if the robot would not initiate avoidance. The participants were instructed to walk with a constant speed towards their goal, matching the velocity of the robot. If they collided with the robot – physical collisions are mitigated by emergency bumpers around the robot – or had to stop to avoid one, it was reported as a collision by the participant. The interaction was started by the participant via a button on a remote control.

For each of the four conditions, we recorded 50 trials in simulation leading to a total of 200 interactions each for pass-by and path crossing. In the proof of concept experiment using the real robot, we recorded two participants, generating 64 pass-by and 61 path crossing situations in total for all 4 conditions combined (with a minimum of 15 each). We evaluated the safety of the trajectory using the number of collisions, the perceived safety by analysing the minimum distance kept to the human, and the efficiency of the executed trajectory in terms of distance travelled, mean speed, and the duration. In the following we will list the results of these two experiments.

A. Results

As can be seen from Tab. II, a high percentage of the generated trajectories lead to the robot colliding with the human, where collisions is to be taken in the literal sense or when the human had to stop to prevent it. We therefore, in simulation, only compare G-Global and the Vel-Maps for pass-by and G-Local and the Vel-Maps for path crossing. All the results of both experiments were generated using an unpaired t-test where *(**)* in Tab. III indicates that all results in this row are highly significant with $p < 0.0001$.

1) Simulation Results: In the pass-by scenario the main difference in results can be seen in the mean minimum distance between the centre of the robot and the centre of the human, denoted *Min Distance* in Tab. III. The absolute difference between the two means is $14cm$ which also results

[6]For reproducibility, the remaining important DWA parameters used were: vx_samples: 3, vth_samples: 20, max_trans_vel: 0.55, max_vel_x: 0.55, max_rot_vel: 1.0, acc_lim_x: 1.0, acc_lim_theta: 3.2, sim_time: 0.8, sim_granularity: 0.025, angular_sim_granularity: 0.1, forward_point_distance: 0.325, scaling_speed: 0.25, max_scaling_factor: 0.2.

[7]Hence, this was the distance at which the robot would start its avoidance manoeuvre.

TABLE II

PERCENTAGE OF TRAJECTORIES COLLIDING WITH THE HUMAN

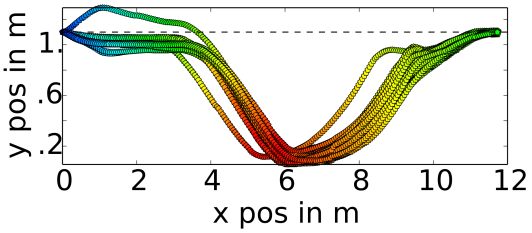
	Simulation		Robot	
	Pass-by	Path Crossing	Pass-by	Path Crossing
DWA	100%	100%	53.3%	86.7%
G-Global	0%	100%	22.2%	75.0%
G-Local	100%	0%	33.3%	100%
Vel-Maps	**0%**	**0%**	**12.5%**	**13.3%**

in a higher travel time, and distance using the Vel-Maps. The absolute difference for the latter, however, is negligible. In the path crossing scenario, the difference in the *Min Distance* amounts to $1.53m$, the difference in distance travelled is only 3cm which implies that both cost functions created straight trajectories (see Fig. 6(b)) like it was shown during the Learning from Demonstration phase.

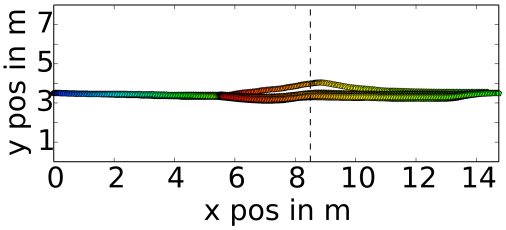
2) Real-World Results: In the path crossing scenario, all approaches but our Vel-Maps resulted in a very high number of collision trajectories. The mean minimum distance to the human for our proposed approach was $0.76m \pm 0.42m$ in path crossing which is considerably lower than in simulation with full observability of the human, but still the highest of all the 4 conditions as can be easily inferred from the number of collisions. For the pass-by scenario the Vel-Maps also achieved the lowest number of collisions. Comparing the two most successful conditions, based on their number of collisions, we measured a mean minimum distances of 0.56m for the Vel-Maps and 0.53m for G-Global with a p value of $p = 0.46$ and therefore no statistical significance. For G-Local, which performed much better on the real robot than in simulation regarding the collisions, we measured a mean minimum distance of 0.47m which with a p value of $p = 0.062$ also comes short of statistical significance when compared to Vel-Maps. Neither mean speed, travelled distance, or duration showed any significant differences between any of the four conditions.

B. Discussion

Our experiments showed that the QTC$_{BC}$ based Vel-Maps approach to human-aware navigation resulted in collision free trajectories in almost all of the cases and shows the behaviour that was trained, i.e. avoiding to the right (see Fig. 6(a)) or stopping to let the human pass (see Fig. 6(b)). By comparing the number of collisions of the Vel-Maps approach for the real-robot and simulation experiment, it becomes apparent that the perfect score of the simulation could not be achieved using a real-robot. This is due to the limitations in the vision component picking up the human too late, due to missing detections, to execute an appropriate avoidance manoeuvre or not at all. If the human was detected at a distance of $> 3m$, collision free trajectories could be generated but the late detection of the human and thereby reduced observability was one of the major downfalls of the real-world experiment. All the conditions suffered equally from this but the Vel-Maps were still able to cope in most of the cases. Given perfect observability in the simulated trials, we showed that only the Vel-Maps were able to prevent collisions in both scenarios. The G-Global cost



(a) Pass-by shows an avoidance movement half way through the interaction when encountering the human.



(b) Path crossing shows an abrupt transition from green to red, visualising where the robot stopped and waited for the human to pass.

Fig. 6. The generated trajectories using Velocity Costmaps in simulation; black dashed line: human trajectory. The robot travelled from left to right and its trajectory is colour coded from blue via red to green to visualise time passed. Robot width: $\sim 60cm$, human width: $\sim 70cm$.

model achieved comparable results in the pass-by scenario, relying on the global path planner, i.e. Dijkstra, to avoid the human, using the local DWA planner only to follow that path and to not collide with walls. The poor performance of the two local obstacle avoidance strategies, i.e. vanilla DWA and G-Local, stems form the DWA planner getting stuck in a local cost maxima and stopping the robot to prevent a collision, despite there not being any additional constraints on the standard DWA. Hence, the bad performance of the DWA in our experiment can be explained by the relatively small planning horizon that in the pass-by encounter has it driving towards the human until it is trapped by the human continuing her approach and in the path crossing by its inability to predict the human motion, thus, driving straight till the human is directly in front of the robot. Given that our human assumed a constant velocity, this stopping behaviour did not prevent collisions in simulation but would have using a real robot, at least in most of the cases as can be seen from the robot trials in Tab. II. Getting stuck in a local cost maxima and stopping is also the reason why the G-Local cost models performed well in the path crossing scenario as they would have the robot stop to let the human pass. The G-Global cost model, however, resulted in the global planner to try and pass in front of the human, leading to collisions because of the relentless motion model used in simulation.

Looking at the mean minimum distance between human and robot in simulation, we see that there is not much difference between the Vel-Maps and G-Global in the pass-by scenario which can be attributed to the size of the corridor itself. The human-robot distance was measured from the centre point of each agent and the human walked in the middle of the 2.6m wide corridor which theoretically leaves

TABLE III

MEAN VALUES FOR SIMULATED SCENARIOS: MIN DISTANCE(MD), MEAN SPEAD(MS), TRAVEL TIME(TT), DISTANCE TRAVELLED(DT). RESULTS WITH (**) ACHIEVED $p < 0.0001$.

	Pass-by			Crossing		
	Vel-Maps		G-Global	Vel-Maps		G-Local
MD (m)	1.06	**	0.92	2.98	**	1.45
MS ($\frac{m}{s}$)	0.52	**	0.53	0.43	**	0.46
TT (s)	23.09	**	22.51	34.11	**	32.03
DT (m)	12.05	**	11.93	14.78	**	14.81

1.3m on either side. In reality this is not achievable without colliding with the wall. The fact that the Vel-Maps approach kept a greater distance is due to the relatively high weight for the human-awareness. In the path crossing scenario, however, we can see that the Vel-Maps approach deliberately restricted the sample space of the DWA to only allow 0 velocities (see Fig. 5(b)) at a much greater distance than all the reactive planners which is an indication for the power and descriptiveness of the QTC$_{BC}$ model. Our robot trials showed that the simulation results are a good indicator for the behaviour shown in real life as all algorithms showed performance comparable to simulation, suffering from the limitations of the human tracker. The lower collision rates could be attributed to the human walking slower than the simulated one and being influenced by his/her sense of self-preservation.

Last but not least, one could argue that the DWA planner using Gaussian cost models is outdated, however, we believe that since it is part of the default ROS navigation stack which is widely used and freely available for a wide range of robot platforms, this can still be considered state-of-the-art.

VI. CONCLUSION AND FUTURE WORK

We presented an approach following a Learning from Demonstration paradigm to select suitable constraints for a local planner, based on a qualitative description language, i.e. QTC$_{BC}$, combined with *Velocity Costmaps* to achieve human-aware navigation by restricting the sample space of a Dynamic Window Approach local planner for obstacle avoidance. Our experiments show, that by encoding high-level knowledge of the unfolding of a possible interaction, we allow our reactive system to cope with a wider variety of possible situations and to make a more informed choice based on the intent of the human. We have also seen that this comes at no extra cost, comparing speed, and travel distance with a standard Gaussian cost approach. This shows that the presented Velocity Costmaps are able to handle the trade-off between safety of the human interaction partner and finding a fast and energy efficient path.

Current work aims at replacing the fixed association of the QTC$_{BC}$ model to a topological edge with a particle filter based classification and prediction of the human's intent. Future work aims at deploying this system in an elder care home for 120 days continuously. This allows to test the proposed Velocity Costmaps in a real-world application and more complex environment. To bootstrap this approach, a database of human trajectories from a previous deployment of 90 days in the same environment will be used.

REFERENCES

[1] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. C. Schultz, and M. Goodrich, "Common metrics for human-robot interaction," *HRI, 2006*, p. 33, 2006.

[2] E. Pacchierotti, H. I. Christensen, and P. Jensfelt, "Evaluation of passing distance for social robots," in *The 15th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2006*, 2006, pp. 315–320.

[3] A. Peters, "Small movements as communicational cues in HRI," in *HRI 2011 - Workshop on Human-Robot Interaction Pioneers*, T. Kollar and A. Weiss, Eds., 2011, pp. 72–73.

[4] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *IEEE International Conference on Robotics and Automation*, vol. 4, no. April. Minneapolis, MN: IEEE, 1996, pp. 3375–3382.

[5] E. Sisbot, L. Marin-Urias, R. Alami, and T. Simeon, "A Human Aware Mobile Robot Motion Planner," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, Oct. 2007.

[6] T. Ducourant, S. Vieilledent, Y. Kerlirzin, and A. Berthoz, "Timing and distance characteristics of interpersonal coordination during locomotion," *Neuroscience Letters*, vol. 389, no. 1, pp. 6–11, Nov. 2005.

[7] C. Dondrup, N. Bellotto, M. Hanheide, K. Eder, and U. Leonards, "A computational model of human-robot spatial interactions based on a qualitative trajectory calculus," *Robotics*, vol. 4, no. 1, pp. 63–102, 2015. [Online]. Available: http://www.mdpi.com/2218-6581/4/1/63

[8] N. Van de Weghe, "Representing and reasoning about moving objects: A qualitative approach," Ph.D. dissertation, Ghent University, 2004.

[9] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[11] A. G. Cohn and J. Renz, "Chapter 13 Qualitative Spatial Representation and Reasoning," in *Handbook of Knowledge Representation*, F. van Harmelen, V. Lifschitz, and B. Porter, Eds. Elsevier, 2008, vol. 3, pp. 551–596.

[12] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889013001048

[13] E. T. Hall, *The hidden dimension*. Anchor Books New York, 1969.

[14] E. A. Martinez-Garcia, O. Akihisa, and S. Yuta, "Crowding and guiding groups of humans by teams of mobile robots," in *Advanced Robotics and its Social Impacts, 2005. IEEE Workshop on*. IEEE, 2005, pp. 91–96.

[15] D. Feil-Seifer and M. Mataric, "People-aware navigation for goal-oriented behavior involving a human partner," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2. IEEE, 2011, pp. 1–6.

[16] L. Scandolo and T. Fraichard, "An anthropomorphic navigation scheme for dynamic scenarios," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 809–814.

[17] D. V. Lu, D. B. Allan, and W. D. Smart, "Tuning cost functions for social navigation," in *Social Robotics*. Springer, 2013, pp. 442–451.

[18] Y. Morales, A. Watanabe, F. Ferreri, J. Even, T. Ikeda, K. Shinozawa, T. Miyashita, and N. Hagita, "Including human factors for planning comfortable paths," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 6153–6159.

[19] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469 – 483, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889008001772

[20] F. Yuan, L. Twardon, and M. Hanheide, "Dynamic path planning adopting human navigation strategies for a domestic mobile robot," in *IROS, 2010.* IEEE, 2010, pp. 3275–3281.

[21] C. Lichtenthäler, A. Peters, S. Griffiths, and A. Kirsch, "Social navigation-identifying robot navigation patterns in a path crossing scenario," *ICSR. Springer*, 2013.

[22] C. Dondrup, N. Bellotto, F. Jovan, and M. Hanheide, "Real-time multisensor people tracking for human-robot spatial interaction," in *Workshop on Machine Learning for Social Robotics*. ICRA/IEEE, 2015.