Real-Time Multisensor People Tracking for Human-Robot Spatial Interaction*

Christian Dondrup[1], Nicola Bellotto[1], Ferdian Jovan[2] and Marc Hanheide[1]

***Abstract*— All currently used mobile robot platforms are able to navigate safely through their environment, avoiding static and dynamic obstacles. However, in human populated environments mere obstacle avoidance is not sufficient to make humans feel comfortable and safe around robots. To this end, a large community is currently producing human-aware navigation approaches to create a more socially acceptable robot behaviour. A major building block for all Human-Robot Spatial Interaction is the ability of detecting and tracking humans in the vicinity of the robot. We present a fully integrated people perception framework, designed to run in real-time on a mobile robot. This framework employs detectors based on laser and RGB-D data and a tracking approach able to fuse multiple detectors using different versions of data association and Kalman filtering. The resulting trajectories are transformed into Qualitative Spatial Relations based on a Qualitative Trajectory Calculus, to learn and classify different encounters using a Hidden Markov Model based representation.**

We present this perception pipeline, which is fully implemented into the Robot Operating System (ROS), in a small proof of concept experiment. All components are readily available for download, and free to use under the MIT license, to researchers in all fields, especially focussing on social interaction learning by providing different kinds of output, i.e. Qualitative Relations and trajectories.

I. INTRODUCTION

Currently used mobile robots are able to navigate safely through their environment, avoiding not only static but also dynamic obstacles. An important aspect of mobile robots however is the ability to navigate and manoeuvre safely around humans [1]. Treating them as dynamic obstacles and merely avoid them is not sufficient in those situations due to the special needs and requirements of humans to feel safe and comfortable when interacting with robots. Human-Robot Spatial Interaction (HRSI), is the study of joint movement of robots and humans through space and the social signals governing these interactions. It is concerned with the investigation of models of the ways humans and robots manage their motions in vicinity to each other. Our work aims to equip a mobile robot with understanding of HRSI situations and enable it to act accordingly.

Recently, the robotics community started to take the dynamic aspects of "human obstacles" into account, e.g. [2] and 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" [3]. The term "legible" here refers to the communicative – or interactive – aspects of movements which previously has widely been ignored in robotics research. To realise any kind of these HRSI applications, the basic challenge is the detection and tracking of humans in the vicinity of the robot considering the robots movement, varying ambient conditions, and occlusion [4]. Due to the importance of these applications, there are several solutions to the problem of detection and tracking of humans or body parts, as shown for example in these surveys [5], [6]. In our work, we are focusing on two specific body part detectors, i.e. human upper bodies and legs, combined in a multisensor Bayesian tracking framework. We present a fully integrated and freely available processing pipeline, using the widely used Robot Operating System (ROS), to detect and track humans using a mobile robot. Apart from the ability to directly feed into reactive human-aware navigation approaches like [7], this detection and tracking framework is used to create qualitative representations of the interaction [8], to facilitate online interaction learning approaches via a Hidden Markov Model (HMM) based representation of these qualitative states.

Detecting walking or standing pedestrians is a widely studied field due to the advances in autonomous cars and robots. To this end, many successful full-body or partly occluded body detectors have been developed, e.g. [9], [10]. Most of these detectors suffer from high computational costs which is why currently used approaches, like the so called *upper body detector* [11], rely on the extraction of a Region Of Interest (ROI) to speed up detection. Therefore, our

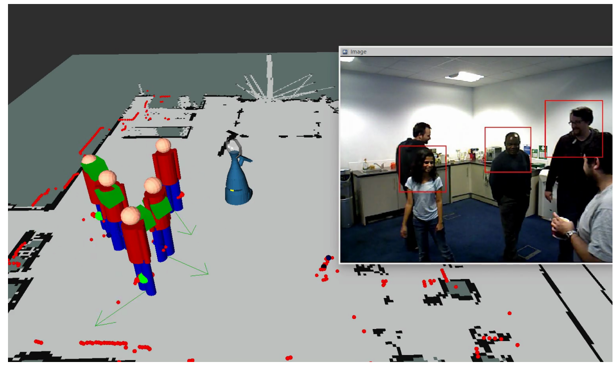


Fig. 1. Example output of the ROS based people perception pipeline, showing the tracker and detector results in `rviz`.

*The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement No. 600623, STRANDS.

[1]Christian Dondrup, Nicola Bellotto and Marc Hanheide are with the School of Computer Science, University of Lincoln, LN6 7TS Lincoln, UK `cdondrup, nbellotto, mhanheide@lincoln.ac.uk`

[2]Ferdian Jovan is with the School of Computer Science, University of Birmingham, B15 2TT Birmingham, UK `fxj345@cs.bham.ac.uk`

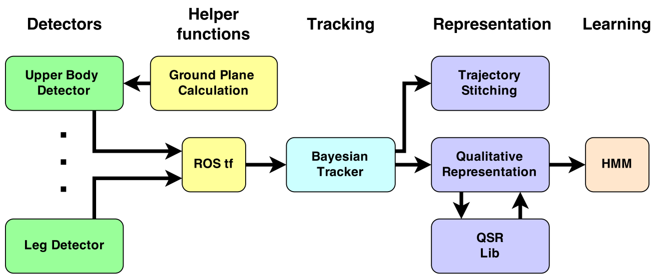


Fig. 2. Conceptual overview of the system architecture. The number of detectors is variable due to the modular design of the tracker and its ability to merge several detections from different sensors.

framework employs this upper body detector for the real-time detection of walking or standing humans. The second detector used is a leg detector based on work by [12] and has become a standard ROS component for people perception. Like detectors, human tracking is an important part of a perception system for human spatial movement. Hence, a variety of tracking systems using multisensor fusion have been introduced, e.g. [13]. We use a probabilistic real-time tracking framework which, in its current implementation of the processing pipeline, relies on the fusion of the two mentioned sensors and an Extended or Unscented Kalman filter to track and predict the movements of humans [14]. However, the tracker itself does not rely on a specific detector for input and is very modular in design.

The presented human perception pipeline (see Fig. 2) is used to facilitate our interaction learning framework which relies on previous work where we introduced a qualitative, probabilistic framework to represent HRSI [8] using a Qualitative Trajectory Calculus (QTC) [15]. We utilised HMMs to represent QTC state chains and classify several different types of HRSI encounters observed from experiments. In this paper we present a fully integrated and automatised processing pipeline for the detection and tracking of people in close vicinity to a robot (see Fig. 1), and the online creation of QTC state chains, enabling the classification and learning of HRSI using our HMM representation. We show the functionality of our framework in a small experiment using an autonomous human-sized mobile robot in a real world office environment, relying on the robots on-board sensors instead of an external motion capture system like in the mentioned previous work.

Everything presented here is available online. A concise list of packages and installation instructions can be found at `http://lcas.lincoln.ac.uk/cdondrup`.

II. OVERVIEW

In this section we present our integrated system, consisting of the perception pipeline including the upper body detector and the tracker, the qualitative spatial representation module, and the library to create the QTC state chains.

A. Detectors

Robots use a range of sensors to perceive the outside world, enabling them to reason about its future state and plan their actions. Our robot (see Fig. 3) used in the experiment has three main sensors, i.e. a Asus Xtion RGB-D camera, a Sick s300 laser, and the odometry of its non-holonomic base. In the following we are presenting the two detectors based on the RGB-D and laser scanner, respectively. Example output can been seen in Figure 1.

Fig. 3. Our Scitos G5 robot – equipped with two RGB-D sensors and a laser scanner – in our open office environment.

The so-called *upper body detector* uses a template and the depth information of a RGB-D sensor to identify upper bodies (shoulders and head), designed to work for close range human detection using head mounted cameras [11]. This first approach was based on stereo outdoor data; an integrated tracking system using a Kinect like RGB-D sensor and the mentioned detector was introduced in [16]. To reduce the computational load, this upper body detector employs a ground plane estimation or calculation to determine a Region of Interest (ROI) most suitable to detect upper bodies of a standing or walking person. The actual depth image is then scaled to various sizes and the template is sliding over the image trying to find matches. This detector works in real time, meaning $\approx 25fps$ which corresponds to the frame rate of the Asus Xtion. We implemented the detector and the ground plane estimation/calculation into ROS and will use it as one of the inputs for our tracker. The main advantage of this detector, compared to full body detectors, is that the camera on our robot is mounted in $1.72m$ hight which only allows it to see upper bodies in normal corridor or room settings, like offices or flats, due to the restrictions in space and the field of view of the Asus camera.

In addition to the RGB-D base detector we also use a laser based leg detector which was initially introduced in [12]. Using laser scanners for people perception is popular in mobile robotics because most currently used platforms provide such a sensor which also has a wider field of view than a camera and is less dependent on ambient conditions. Arras *et al.* define a set of 14 features for the detection of legs which uses, e.g. the number of beams, the circularity, the radius, mean curvature, and the mean speed, to only name a few. These features are used for the supervised learning

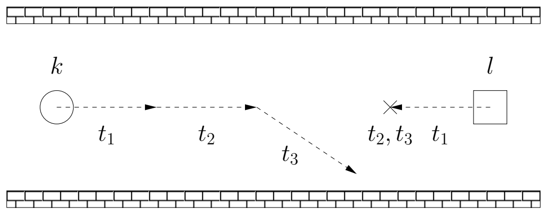


Fig. 4. Example of HRSI encoded in QTC, with human k and robot l. The respective QTC state chain is $(-\ -\ 0\ 0)_{t_1}$: k moves straight towards l $(\mathbf{-}\ -\ \mathbf{0}\ 0)$ and l moves straight towards k $(-\ \mathbf{-}\ 0\ \mathbf{0})$, $(-\ 0\ 0\ 0)_{t_2}$: k continues its movement while l is stationary $(-\ \mathbf{0}\ 0\ 0)$, $(-\ -\ +\ 0)_{t_3}$: k approaches and moves to the right $(\mathbf{-}\ 0\ \mathbf{+}\ 0)$ while l stays stationary. For a more comprehensive and detailed description of QTC for HRSI please refer to [8].

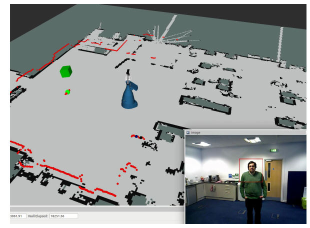
of a set of weak classifiers using recorded training data. The AdaBoost [17] algorithm is employed to turn these weak classifiers into a strong classifier, detecting legs in laser range data. The approach was evaluate in various office and corridor settings which makes it ideal for most indoor robotics environments. The implementation of the detector is part of the official ROS people stack[1] and is also used to feed into our tracker.

B. Tracker

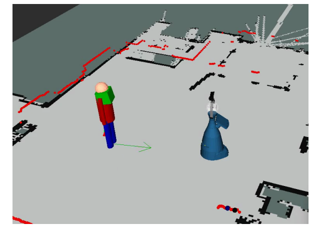
To use the wealth of information provided by a robot equipped with multiple sensors, we refrain from using purely vision based trackers like the one introduced by Hosseini *et al.*[2] – from which we extracted the upper body detector – and employ a solution for *Bayesian tracking* originally implemented in [14]. This tracker is available in its current implementation from [18] (see Fig. 5(b)) and allows to natively combine multiple sensors and is not dependent on the frame rate of any one detector. Bellotto *et al.* showed that their Bayesian tracker, based on an Unscented Kalman Filter, achieves comparable results to a Sampling Importance Resampling particle filter in several people tracking scenarios, although it is computationally more efficient in terms of estimation time. In the current implementation different tracking configurations can be used by defining the noise parameters of the constant velocity model to predict human motion, to e.g. compensate for loss of detection, as presented in [19], and the fixed frame observation models (one for each detector). A gating procedure is applied using a validation region relative to the target, based on the chosen noise parameters, for each new predicted observation in order to reduce the chance of assigning false positives and wrong observations [20]. New detections are then associated to the correct target using a Nearest Neighbour (NN) association algorithm, suitable for computationally less powerful robot systems, or a more sophisticated Nearest Neighbour Joint Probabilistic Data Association (NNJPDA), which is more reliable but also less efficient regarding computation time. If no suitable target could be found, the detections are stored

[1] https://github.com/wg-perception/people

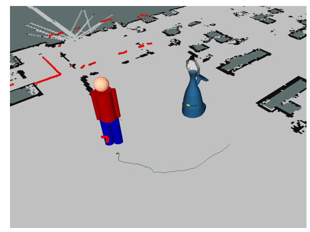
[2] The tracker bey Hosseini *et al.* has been ported and split up into several ROS modules and is available from our github repository or debian package server, see Sec. II-E.



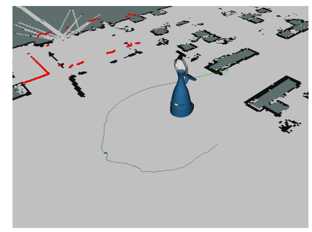
(a) The two used detectors. Green sphere: leg detector, green cube and red box in image: upper body detector.



(b) The tracker output. Overlaid red and blue figure: position of tracked human, green arrow: orientation.



(c) A human moving around the robot. Showing his/her current position and the path since the start of the tracking.



(d) The complete path of the human walking around the robot. Human left the field of view of the laser and is not tracked any more.

Fig. 5. The visualisations of the detector and tracker outputs using rviz. The leg detector is a standard ROS component, the upper body detector was ported to ROS and tracker was implemented by us.

and eventually used to create a new track if they are stable over a predefined time frame. For a detailed description of the tracking and association algorithms, please refer to the original paper [14] and other relevant work [21].

C. Spatial Interaction Learning

The output of the tracking framework, i.e. the position, velocity, and orientation of the tracked humans, can either be directly used for reactive human-aware navigation like the ROS implementation of layered costmaps [7], or for online spatial interaction learning. In previous work, we focused on a learning approach representing HRSI utilising Qualitative Spatial Relations (QSR) [8], i.e. a Qualitative Trajectory Calculus (QTC) [15]. This is used to qualitatively represent the relative spatial movement of a human (k) and a robot (l) in relation to one another (see Fig. 4).[3] The 2D movement of these two agents k and l is represented in a 4-tuple $(a\,b\,c\,d)$ where a and b can assume the values $\{-,+,0\}$ for moving apart, towards the other, or being stable with regards to the last position and where c and d can assume the values $\{-,+,0\}$ for moving to the left, right, or along the connecting line between the two agents. Hence, a and c describe the movement of k and b and d the movement of l in relation to each other. This allows to abstract from metric space and to focus on the "essence" of the interaction.

We presented a Hidden Markov Model (HMM) representation of these QTC state chains in [8] used to classify and reason about HRSI situations. This probabilistic model enables us to represent actual sensor data by allowing for

[3] The variant of QTC we are referring to here is QTC_{C21}.

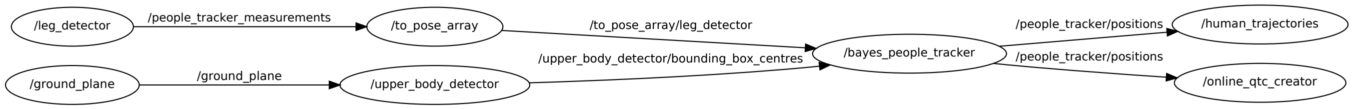


Fig. 6. A simplified representation of the ROS nodes and their most important connections, created using `rqt_graph`.

uncertainty in the recognition process. We were able to reliably classify different HRSI encounters, e. g. pass-by and overtake scenarios, and showed that the QTC-based representations of these scenarios are significantly different from each other [8]. We have initially modelled the "correct" emissions, e.g. $(+ - 0\ 0)$ actually emits $(+ - 0\ 0)$, to occur with 95% probability and to allow the model to account for detection errors with 5%. Our HMM contains the legal transitions stemming from QTC and the transitions from and to the start and end state, respectively. For each different behaviour to be represented, a separate HMM is trained, using Baum-Welch training [22] (Expectation Maximisation) to obtain the appropriate transition and emission probabilities for the respective behaviour. In the initial pre-training model, the transitions that are *valid*, according to the Continuous Neighbourhood Diagram for QTC [15], which defines the legal transitions between states, are modelled as equally probable (uniform distribution). We allow for pseudo-transitions with a probability of $P_{pt} = 1e^{-10}$ to overcome the problem of a lack of sufficient amounts of training data and unobserved transitions therein. These HMMs can then be used to classify the observed interactions and to predict their outcome by sampling from the most probable paths.

D. Trajectory Stitching

In order to enable and facilitate more generic trajectory learning approaches, in addition to our Qualitative Spatial Relation based framework, we generate trajectories of human and robot, accounting for periods of occlusion or loss of detection in general. We regard a human trajectory as a sequence of positions stitched together based on the chronology of the poses. The trajectories of all agents are taken into account, i.e. the trajectory information of all the currently tracked humans and the robots trajectory including the length of the human trajectories, and the start and end time. To create these, all human poses are grouped based on the Universally Unique Identifier (UUID) provided by the tracker. Each UUID represents a person and once there is no position update for that particular UUID, a validation of the recorded poses is carried out. The validation includes ordering the poses based on time of appearance and taking one pose out of k-similar poses recorded at the same timestamp. We selected the pose with the smallest distance between the next and the previous pose.[4] Once the validation has taken place, the human trajectory is ready to be stored in a database and published in form of a ROS message. For the online stitching, human trajectories are published incrementally (see Fig. 5(c) and 5(d)), meaning that the human trajectories will be published in a chunk of incomplete trajectories. The sequence id of the published message states the order of the chunks so that by combining all the published messages, a complete human trajectory can be obtained.

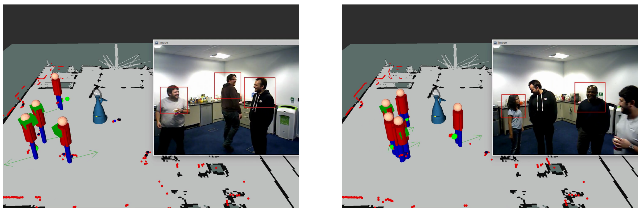
[4]In the presented framework, this step is obsolete and just mentioned for the sake of completeness and to show the appropriateness of this particular component for noisy detectors or trackers.

E. System Dependencies, Installation, and Usage

This work presents the implementation of all the mentioned components into a processing pipeline (see Fig. 2), able to work in real-time (prediction rate $> 30Hz$) on a mobile robot. It has been tested on a Scitos G5 robot (see Fig. 3), using 2 embedded PCs, i.e. an i5 with 4GB RAM for the navigation and hardware communication and an i7 with 16GB RAM for the image processing. No dedicated graphics cards are required. The robot is equipped with two Asus Xtion RGB-D sensors, one mounted overhead on a Pan Tilt Unit for the upper body detector, the other at chest hight facing down for 3D obstacle avoidance, and a Sick s300 laser for navigation and leg detection. Software dependencies are Ubuntu 12.04 or 14.04 64bit, and the Robot Operating System (ROS) (supported versions: hydro - maintained and indigo - developed), see Figure 6. All software is distributed under the MIT license where possible and can be downloaded from our github page or installed via debian packages from our private server, please refer to the website `http://lcas.lincoln.ac.uk/cdondrup`. The original pipeline relies on a RGB-D sensor and a laser scanner but the tracker's modular design allows for the easy replacement or addition of detectors. To add a detector to the tracking framework, we have to provide the following information via a YAML based config file:

```
bayes_people_tracker:
  filter_type: 'UKF'
  cv_noise_params:
    x: 1.4
    y: 1.4
  detectors:
    my_detector:
      topic: '/my_topic'
      cartesian_noise_params:
        x: 0.5
        y: 0.5
      matching_algorithm: 'NNJPDA'
```

where `filter_type` can be `UKF` or `EKF` for the Unscented and Extended Kalman Filter, respectively. `cv_noise_params` describes the standard deviation of the noise (i.e. acceleration component of the motion) for the constant velocity prediction model. Please note, that a standard Kalman Filter would be sufficient for the use with such a model but the tracking library itself [18] allows for



(a) Four people are tracked. Two of them via the upper body detector, one via the leg detector, and one via the combination of both using NNJPDA.

(b) Five people are tracked. The upper body detector only picks up two due to occlusion or "incorrect" body posture. The other two are tracked via the leg detector input.

Fig. 7. As a proof of concept, we tracked several people moving around our office environment, showing that the multisensor tracking compensates for false negatives of the detectors.

the easy addition of other, non-linear predication models. Under the `detectors` namespace we can add any arbitrary amount of detectors defined by: `my_detector` as a unique identifier, `topic` is the ROS topic under which the geometry_msgs/PoseArray for the detections is published, the `cartesian_noise_params` describing the standard deviations of the noise for the Cartesian observation model, and the `association_algorithm` which can either be `NN` or `NNJPDA` for the previously presented data association approaches. All noise parameters just represent example values which work for the presented evaluation but could of course be enhance by empirical analysis of the sensor noise. We show the usage of the whole system in a video created during the evaluation described in Section III and linked on our website.

III. EVALUATION

The majority of the presented components have been evaluated separately in previous publications. For the evaluation of the detectors, please refer to [11] and [12], respectively. An exhaustive evaluation of the tracker can be found in [14] and the QTC based HMM representation of HRSI has been presented and evaluated in [8].

Since this work presents the integration of all these components into a state-of-the-art robot platform and the widely used Robot Operating System, we are presenting a short proof of concept. To this end, we deployed the robot in our open office environment, observing people in the kitchen area, see Figures 1 and 7. A video of this deployment is available on youtube, please refer to our website for the link. Screenshots of the live system can be seen in Figures 1, 5, and 7.

In addition to the proof of concept for the perception pipeline, we conducted a short trial using the tracker output to generate QTC state chains online; feeding them into our HMM based representation. The trial featured our robot driving along a corridor and an oncoming human, engaging in a so-called *pass-by* encounter. The robot would drive straight while the human was circumventing the robot either on the left or right side. This was repeated 7 times for each side. We classified the resulting state chains into passing on the left vs. passing on the right using the dataset presented in [8], featuring two human interactants. The HMMs for classification were trained using the trajectories of both humans recorded via a motion capture system.

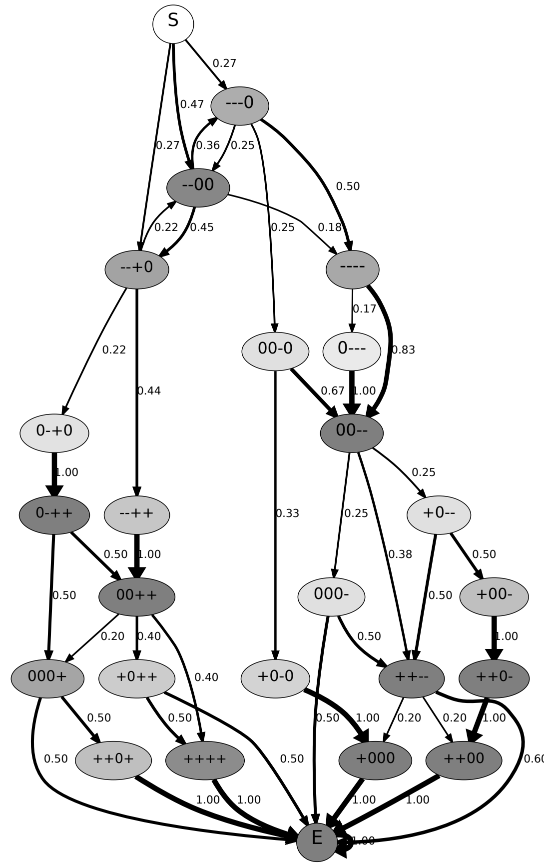


Fig. 8. Visualisation of the HMM trained from the recorded QTC state chains. A $-$ in the 3rd and/or 4th position of the tuple indicates circumvention on the left and a $+$ in the respective positions represents circumvention on the right. Transitions below 0.15 have been pruned for visualisation purposes. The grey level represents the a-priori probability of the state, the darker the higher the probability of being in this specific state.

A. Results

Figure 8 shows the recorded QTC state chains in our HMM based representation. The figure is clearly divided into two possible paths, passing on the left and passing on the right as can be seen from the $-$ or $+$ for c and d in the tuple $(a\,b\,c\,d)$.

As stated above, we classified our recorded encounters using models trained from data collected during a Human-Human Spatial Interaction experiment using motion capture. The classifiers were trained for three different conditions, i.e. starting the circumvention early, late or in between where each is $500ms$ apart. Each of the six models (three per side) was trained with 162 to 178 (for passing on the right) and 183 to 189 (for passing on the left) QTC state chains, respectively. Using these models and the state chains generated using sensor data via our tracking framework, we achieved classification rates from 78.57% to 85.71% on our dataset using input generated from our integrated systems instead of external motion capture. In the following we are discussing these results and the general implementation of the perception pipeline.

IV. Discussion

This work presents components that have already been proposed for human detection, tracking, the generation of qualitative representations of HRSI, and trajectory stitching. However, for the first time, all these components have been implemented in a concise and easy to deploy system based on ROS the Robot Operating System. The majority of these components have therefore already been evaluated on their own which makes it hard to argue for the novelty of the system. Nevertheless, we believe that the presented people perception pipeline, especially considering the automated QSR and trajectory generation, provides a specialised tool set facilitating offline and online learning approaches for Human-Robot Spatial Interaction. Our short trial indicates that, using the automatically generated QRSs, we are able to classify HRSI encounters using models created from previous knowledge, based on a similar scenario but recorded using a vastly different sensor and modelling the interaction between two humans. We believe that this ability of knowledge transfer from previous encounters or even from Human-Human Spatial Interaction, is a very promising and interesting direction of research and shows the power of our tracking framework and QSR based model.

Limitations and Lessons Learned: Like all robotic systems, our approach is very susceptible to sensor noise and limited fields of view when it comes to detectors. The upper body detector itself for example has shortcomings when it comes to detecting sitting people due to the defined Region of Interest being optimised for walking/standing people, and the shape of the used depth template. This template represents an upright upper body, which can mostly be found in walking or standing persons. Looking at Figure 7(b) we can see that one of the people in the images is not detected due to an "incorrect" body posture. The solution to most of these issues is provided by the Bayesian tracker, fusing the detectors to compensate for the fact that one of them might produce false negatives or a person is not in its field of view, and via smoothing the trajectories using a Kalman filter and a constant velocity model.

Concluding from the above statements, we did not specifically focus on the development of novel approaches to solve human-aware or social navigation but on providing a toolbox to facilitate spatial interaction learning, making it readily available to be employed for existing reactive human-aware navigation or more sophisticated machine learning techniques; one of them being the presented QSR based probabilistic model.

Acknowledgment

The authors would like to thank Dennis Mitzel for his help in "ROSifying" the upper body detector and Yiannis Gatsoulis for implementing the basic functionalities of the QSR library.

References

[1] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. C. Schultz, and M. Goodrich, "Common metrics for human-robot interaction," *Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction - HRI '06*, p. 33, 2006.

[2] 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.

[3] 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.

[4] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 143–166, mar 2003.

[5] G. S. Walia and R. Kapoor, "Human detection in video and images - a state-of-the-art survey," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 03, 2014.

[6] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer vision and image understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[7] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 709–715.

[8] 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

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[10] C. Wojek, S. Walk, S. Roth, and B. Schiele, "Monocular 3d scene understanding with explicit occlusion reasoning," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1993–2000.

[11] D. Mitzel and B. Leibe, "Close-range human detection for head-mounted cameras," in *British Machine Vision Conference (BMVC)*, 2012.

[12] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3402–3407.

[13] S. Feyrer and A. Zell, "Robust real-time pursuit of persons with a mobile robot using multisensor fusion," in *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, 2000, pp. 710–715.

[14] N. Bellotto and H. Hu, "Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of bayesian filters," *Autonomous Robots*, vol. 28, no. 4, pp. 425–438, 2010.

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

[16] O. H. Jaffari, D. Mitzel, and B. Leibe, "Real-Time RGB-D based People Detection and Tracking for Mobile Robots and Head-Worn Cameras," in *IEEE International Conference on Robotics and Automation (ICRA'14)*, 2014.

[17] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine learning*, vol. 37, no. 3, pp. 297–336, 1999.

[18] N. Bellotto, C. Dondrup, and M. Hanheide, "bayestracking: The bayes tracking library v1.0.5," Mar. 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.15825

[19] X. R. Li and V. P. Jilkov, "A survey of maneuvering target tracking: Dynamic models," in *Proc. of SPIE Conf. on Signal and Data Processing of Small Targets*, FL, USA, 2000, pp. 212–235.

[20] Y. Bar-Shalom and X.-R. Li, "Multitarget-multisensor tracking: principles and techniques," *Storrs, CT: University of Connecticut, 1995.*, 1995.

[21] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009.

[22] G. A. Fink, *Markov Models for Pattern Recognition*. Springer-Verlag Berlin Heidelberg, 2008.