Interactive Map Labeling for Service Robots

Alexander J. B. Trevor, Akansel Cosgun, Jayasree Kumar, Henrik I. Christensen

***Abstract*— Maps that include semantic information such as labeled structures, objects, or landmarks can be useful to service robotic systems. For example, we might want our service robot to be able to accept commands such as "fetch the mug from the kitchen table". To perform such a task, the robot should know the location and extent of the structure referred to by "kitchen table" – that is, we must establish common ground for this term. We describe a map representation capable of representing such labeled structures, and an interactive annotation system for generating labeled maps.**

I. INTRODUCTION

Maps that include semantic information such as labels for structures, objects, or landmarks can provide context for navigation and planning of tasks, and facilitate interaction with humans. For example, we might want our service robot to be able to accept commands such as "fetch the mug from the kitchen table". To perform such a task, the robot should know the location and extent of the structure referred to by "kitchen table". As another example, we might want the robot to "go down the hallway". To do this, it is useful to have a representation of the location and extent of the hallway. In this way, we must establish "common ground" [3] for communication, by grounding these labels to landmarks in the robot's map.

A simple approach to adding labels to robot maps would be to attach labels to specific coordinates in the map. For example, we might attach a label "kitchen table" to coordinate $(2.4, 7.12, 0.0)$. Such an approach fails to capture the shape and extent of the structure designated by the label, which might be important to tasks such as object search. In addition, point based references may be ambiguous for a region or volume in a map, such as a hallway or room. For this reason, we will take an approach that allows us to represent both location and extent of landmarks and spaces corresponding to these labels.

We propose a map representation that includes planar patches represented by a plane equation in hessian normal form, a polygon representing the boundary of the surface, and an optional label. Surfaces can be detected from point cloud data, and represented in a consistent map coordinate frame. Maps composed of such features can represent the locations and extents of landmarks such as walls, tables, shelves, and counters. This need not be the only information in the map – occupancy information or other landmark types could also be mapped and used for other purposes such as localization or obstacle avoidance. We employ a SLAM system described in previous work [24] that uses such surfaces as landmarks in a feature-based SLAM system.

Many task-relevant landmarks may be represented as single planar structures that have unique labels, such as "coffee table" or "front door". Other labels might correspond to regions of a map bounded by multiple planar landmarks, such as the walls of a room or hallway. If a user labels multiple planar landmarks with the same label, such as the four walls of a room, or two walls of a hallway, it specifies a region of space corresponding to this label by finding the convex hull of the extent of all planar features with that label. In this way, we can label both specific landmarks such as tables or shelves, as well as regions such as rooms, hallways, or cubicles.

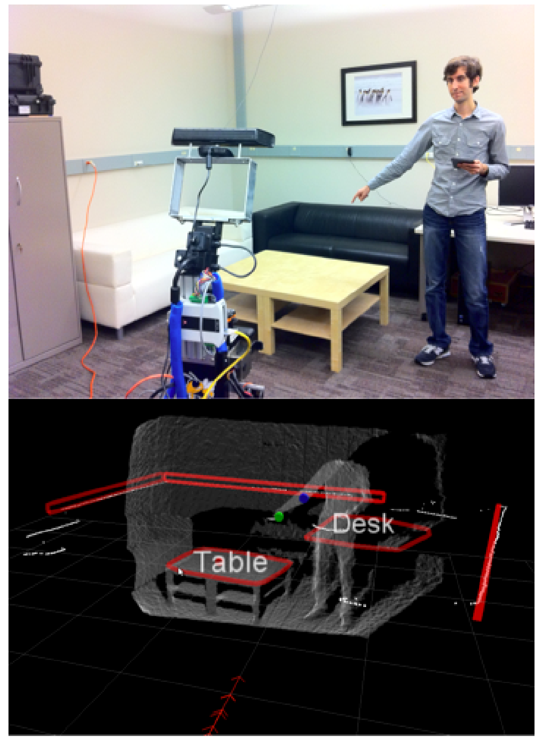

Fig. 1. Top: A photo of a user pointing at a nearby table, after entering the desired label on a handheld tablet UI. Bottom: A visualization of the robot's map and sensor data. Red outlines represent the convex hulls of mapped planar features. The detected gesture can be seen via the green sphere at the user's hand, the blue sphere at the user's elbow, and the red sphere (near the "a" in Table) indicating where the pointing gesture intersects a mapped plane.

To label the map, we have developed an interactive system that uses a combination of a tablet based user interface and pointing gestures. Using this tablet interface, users can command the robot to follow them through a mapped environment, as well as enter labels for nearby features. Labels are entered using an on-screen keyboard. Pointing gestures are recognized using data from a Microsoft Kinect, and the referenced landmark is annotated with the entered label. An example of the labeling process, as well as a map generated by our system is shown in Figure 1.

A brief survey of related work will be given in Section II, followed by a detailed description of our approach in Section III. Results are presented in Section IV, followed by a conclusion in Section V.

II. Related Work

There are several areas of related research. The most closely related approach to our own is that of Human Augmented Mapping (HAM), introduced by Topp and Christensen in [23] and [22]. The Human Augmented Mapping approach is to have a human assist the robot in the mapping process, and add semantic information to the map. The proposed scenario is to have a human guide a robot on a tour of an indoor environment, adding relevant labels to the map throughout the tour. The HAM approach involves labeling two types of entities: regions, and locations. Regions are meant to represent areas such as rooms or hallways, and serve as containers for locations. Locations are meant to represent specific important places in the environment, such as a position at which a robot should perform a task. This approach was applied to the Cosy Explorer system, described in [27], which includes a semantic mapping system that multi-layered maps, including a metric feature based map, a topological map, as well as detected objects. While the goal of our approach is similar, we use a significantly different map representation, method of labeling, and interaction.

Another related body of work is that of semantic mapping, which aims to create maps that include various types of semantic information to allow the robot to have a richer representation of the environment. Many approaches have been reported, some of the key efforts include the Spatial Semantic Hierarchy (SSH) of Kuipers [11], the semantic mapping work of Martínez-Mozos and Rottmann [13] [17], as well as the place categorization of Pronobis *et. al* [16].

There are also several efforts directly related to our mapping approach. Our mapper uses a smoothing and mapping approach based on the work of Dellaert [4], which we use with our own planar landmark mapping technique. Others have also proposed similar solutions to using planar features for mapping, including Weingarten [25], Pathak *et.al.* [15], and Rusu *et. al.* [19].

The problem of following a person with a mobile robot has found a lot of interest in the literature. One of the methods to track people is to detect legs in laser scans [1], [21]. One other common method is to use face or blob detection and fuse it with laser-based methods [9]. More recently, RGB-D cameras have been used to detect and follow people. Loper [12] presents a system that follows a person and recognizes speech commands as wells as non-verbal gestures. In that work, specifically designed arm gestures were used to start/stop following and give lower level motion commands to the robot. There also has been work to recognize natural gestures, such as the gesture of pointing to a direction. Nickel [14] estimated the pointing direction by detecting the head and hand in a stereo image. Steifelhagen [20] detected pointing gestures and used it in the context of Human-Robot Interaction. Fong [7] describes a robot that can be controlled by both arm gestures and a handheld device. Dias [6] have used a multi-modal interface combining speech and gesture in a tablet computer for a multi robot - multi user setting. Using touch screen devices as the main User Interface(UI) is also common for robots [8].

III. Approach

Our map annotation approach requires three main components: a mapping system, a person following system, and a label annotation user interface. The system diagram shown in Figure 2 demonstrates the relationship between these components. For mapping, we use a SLAM system from the authors' previous work described in [24]. For the map annotation, we utilize a combination of a tablet user interface and gesture recognition. The expected usage scenario is similar to that of the Human Augmented Mapping [22] approach, where a user can command the robot to follow them on a tour of an environment, and stop to annotate important landmarks and regions.

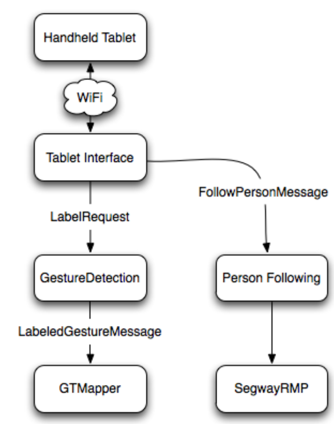

Fig. 2. A system diagram of our approach.

A. Map Representation

Many robotic mapping systems employ occupancy-based information. While this can be helpful for localization and path planning purposes, maps may also need to contain other types of spatial information for other applications. Our approach relies on keeping track of a set of observed planar surface as part of the map representation. Planes are represented in the hessian normal form by the well known equation: $ax + by + cz + d = 0$. Since the observed planes do not have infinite extent, we bound the observed area with a polygon – in our case, the convex hull of all observed points on the plane. We allow these planar landmarks to have an optional label. This results in a map that contains a list of planes of the form $p = \left[n, boundary, label\right]$, where $n = \left[a, b, c, d\right]$ and $boundary$ is a polygon that bounds the observed planar surface.

B. Mapping System

Our mapping system takes a feature-based approach, and supports a variety of feature types that can be used for SLAM and semantic mapping. A detailed description of the mapping system is presented in [24]. In this work, we will only consider planar features. These landmarks can be measured either via sensors that detect point cloud data, or 2D measurements can be made on them using a 2D laser scanner. Planar measurements only measured by 2D sensors are assumed to be vertical.

3D laser scanners or RGB-D sensors such as the Microsoft Kinect can be used to collect suitable data. Planes are extracted from the point cloud by an iterative RANdom SAmple Consensus (RANSAC) method, which allows us to find all planes meeting constraints for size and number of inliers. A clustering step is performed on extracted planes to separate multiple coplanar surfaces, such as two tables with the same height, but at different locations. We make use of the Point Cloud Library (PCL) [18] for much of our point cloud processing.

Planes can be represented in the hessian normal form: $ax + by + cz + d = 0$ and the vertices of the plane's convex hull. As the robot platform moves through the environment and measures planes multiple times, the mapped planes' hulls will be extended with each new portion seen, allowing the use of large planes such as walls where the full extent is typically not observed in any single measurement.

The optimization engine used in our mapping system is the Georgia Tech Smoothing and Mapping library (GTSAM) [4]. GTSAM solves the smoothing and mapping problem using factor graphs that relate landmark poses to robot poses. The factors are nonlinear measurements produced by measuring various features. New factor types can be defined and used by specifying a measurement function along with its derivatives with respect to the robot pose and the landmark pose. The resulting factor graph can be used to optimize the positions of the landmarks as well as the robot poses. Note that we are solving the *full SLAM* problem, recovering not just the current robot pose, but the full trajectory along with the landmark poses.

C. Gesture Recognition

We use pointing gestures as a natural interface between the robot and the user. The pointing direction is used to determine which planar surface the user is pointing to. To detect people in the environment, we used a Microsoft Kinect RGB-D sensor with OpenNI natural interaction framework for skeleton tracking. OpenNI necessitates the person to make a calibration pose to start skeleton tracking. This limitation actually serves to our purposes since we allow only one user to be in control and the person who does the calibration pose is chosen to be the user. Pointing gesture detection is initiated when a LabelRequest message is received, which contains the label entered by the user. The gesture is then searched for T seconds. Let's denote ${}_se$ and ${}_sh$ as the elbow and hand position of one arm in the sensor's coordinate frame. Let ϕ_g denote a threshold angle above which the user's arm must be

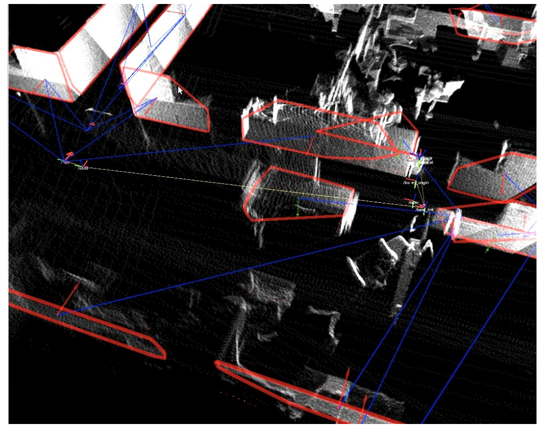

Fig. 3. An example of a map generated by our system. The map shown is of an office environment. The map includes planar surfaces such as walls, tables, and desks, as shown by the planar surfaces outlined by the red convex hulls. Surface normals are also displayed in red, on the centroids of the landmarks' hulls. The robot's trajectory is shown as a sequence of red arrows, and blue lines between trajectory poses and landmarks indicate the landmarks measured from that pose. The point clouds used to construct the map have been registered in the map frame, and are displayed in white, for visualization purposes only – full point clouds are normally not needed.

raised to be considered a pointing gesture. Two conditions have to be satisfied to trigger a pointing gesture:

1) $\frac{{}_se_z - {}_sh_z}{|{}_se - {}_sh|} > sin(\phi_g)$
2) ${}_se$ and ${}_sh$ stays almost constant for some duration Δt_g.

The first condition requires the forearm not to be on the side of the body, and the second ensures that the gesture is consistent for some time period. We have used $T = 30s$, $\phi_g = -45\,^\circ$ and $\Delta t_g = 0.5s$. Whenever the pointing gesture is triggered, a LabeledGestureMessage is sent to the mapper. The mapper then applies the coordinate transformations ${}_me = ({}_mT_s){}_se$ and ${}_mh = ({}_mT_s){}_sh$, which allows us to find all the planes that intersects the ray emanating from ${}_me$ towards ${}_mh$ in the map frame. When testing if the ray intersects with a given plane, we also enforce that the intersection point lies within the convex hull of that landmark. If the ray intersects with multiple planar landmarks, the closest one to the user's hand is selected.

D. Person Following

Person tracking is achieved by detecting legs in the laser data acquired from a laser scanner placed parallel to floor at ankle height. First, the laser scan is segmented. Two adjacent distance measurements are considered to be in the same segment if the euclidian distance between them is below a threshold. Then the segments are scored according to their likelihood to be a leg. In a laser scan, legs can appear in different patterns [21]. We track only one *Single Leg(SL)* pattern. Three geometric features of a segment are calculated for leg detection: Segment Width, Circularity and Inscribed Angle Variance(IAV)[26]. We captured several laser frames while both the robot and the person are mobile and manually labelled about 1.7×10^4 *SL* patterns. For the training set, two people's legs are recorded with different clothing (i.e. shorts, baggy pants, trousers). Mean and variances of leg features are calculated and are used as constants for leg detection.

For a segment, the leg features are calculated and the weighted mahalanobis distance to the mean leg features is calculated. If the mahalanobis distance is below a threshold, the segment is identified as a leg. While the leg is being tracked; a fourth parameter, the proximity of the segment center to the estimated position of the leg, is also considered. The leg is tracked with a Kalman Filter in the odometry frame using a constant velocity model. When no segment is below the leg distance threshold, the tracker expects the leg to reappear in the same location for some time, before declaring the person lost. This allows handling of no-detection frames and temporary occlusions(i.e. another person passes between the tracked person and robot). The mean and variances of leg features are adaptively updated using the last 20 frames, so that the tracker adjusts to the leg shape/clothing of the person after the initial detection. After the command to follow a user is received, the person to follow is selected by detecting a significant movement by a leg segment in the region in front of the robot.

The leg position is considered as the position of the person and the robot's objective is set to be 0.8m away from the person and oriented towards him/her. The goal position is re-calculated at every laser scan and provided to a local navigation planner. We have used the Robot Operating System (ROS) navigation stack to implement the person following behavior.

E. Tablet UI

A tablet-based interface was developed for interaction between the user and the robot. Similar functionality could be achieved using a speech dialog system, but such interfaces tend to have a high error rate [5], and ultimately we decided to use a the tablet as a more robust way of communicating with the robot. The interface has been implemented on a Samsung Galaxy Tablet running the Android OS. The tablet is equipped with 802.11 WiFi, and can communicate with a ROS node running on our robot via a TCP socket.

The interface contains a series of buttons which can be used via the tablet's touch screen. A screenshot of the tablet's UI is shown in Figure 4. This interface can be used to command the robot to follow the user, stop following the user, annotate landmarks with labels, and navigate to labeled landmarks. Labels are entered via the tablet's on-screen keyboard, and sent to the robot.

F. Navigating to labeled structures

After labeling various landmarks in the map, we want the robot to be able to navigate back to labeled landmarks. We have implemented a function to calculate a goal location corresponding to a label.

We consider two cases when calculating a goal point for a given label. The requested label may correspond to one landmark, or multiple landmarks. In the case that it is one landmark, it corresponds to a plane somewhere in the environment, but if many landmarks share the label, it corresponds to a volume.

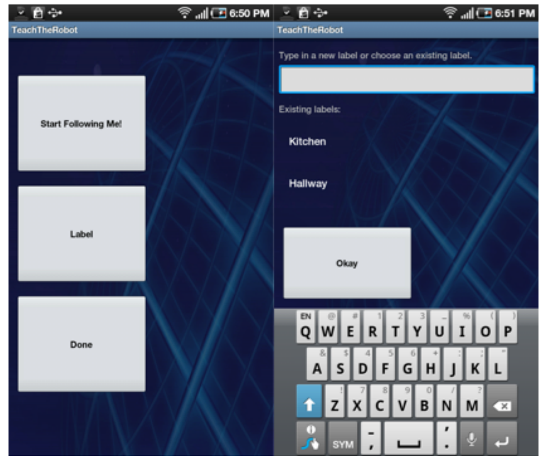

Fig. 4. Screenshots of our tablet user interface. Left: A screenshot of the interface that allows the user to command the robot to follow, or stop following them, or enter a label. Right: A screenshot of the interface for labeling landmarks. Previous labels are available for the user's convenience when labeling multiple surfaces with the same label (e.g. the four walls of a room). New labels can be entered via the onscreen keyboard. Upon pressing the "Okay" button, the user points at the landmark that should be annotated with the desired label.

Let us first consider the case where there is only one plane with a given label. In this case, we assume that the robot should navigate to the closest edge of the plane, so we select the closest vertex on the landmark's boundary to the robot's current position. This point is projected down to the ground plane, as our robot navigates on the floor. We calculate a line between this point and the robot's current pose, and navigate to a point on this line 1 meter away from the point, and facing this point. This results in the robot navigating to near the desired landmark, and facing it. This method is suitable for both horizontal planes such as tables, or vertical planes such as doors.

In the case that we have multiple planar landmarks annotated with the same label, this label corresponds to a region of space such as a room or corridor. In this case, we project the points of all planes with this label to the ground plane, and compute the convex hull. For the purposes of navigating to this label, we simply navigate to the centroid of the hull. While navigating to a labeled region is a simple task, this labeled region could also be helpful in the context of more complex tasks, such as specifying a finite region of the map for an object search task.

IV. Results

Preliminary validation of the system has been performed on a robot platform in our lab and office environment. The environment was mapped *a priori* by teleoperating the robot through the environment. The interactive labeling system was then demonstrated on the robot using this map. A formal user study of the system with non-expert users is future work.

A. Robot Platform

The robot platform used in this work is the *Jeeves* robot from the Georgia Institute of Technology's Cognitive

Robotics Lab, shown in Figure 6 and 8. Jeeves is comprised of a Segway RMP-200 mobile base, and is equipped with a variety of sensors. A SICK LMS-291 laser scanner is used for navigation and obstacle avoidance. A Microsoft Kinect RGB-D sensor is mounted on a Directed Perception DP-47-70 pan-tilt unit, allowing us to capture point cloud data, which we require for mapping. The platform is also equipped with a parallel jaw gripper mounted on a 1-DOF linear actuator, which allows basic manipulation tasks when combined with the Segway's additional degrees of freedom.

B. Labeling System

We used our system in an office environment for two types of scenarios. First, labeling single important landmarks, such as the coffee table and desk , seen in Figure 1. Once a metric map was collected, the user used the tablet-based UI to command the robot to follow them through the environment to important landmarks. The tablet UI was used to type the relevant labels for the landmarks, and gestures were used to indicate which landmark should be annotated with this label.

We also tested labeling of regions by labeling multiple surfaces with the same label. For example, a hallway can be seen in Figure 7.

C. Navigating to labeled landmarks and regions

Navigation to labeled landmarks has also been tested by using the tablet-based user interface. An example goal location corresponding to the labeled table is shown in Figure 5. As was described in Section III-F, the goal location is calculated by first selecting the nearest point on the landmark's convex hull to the robot, as is shown as the white sphere in Figure 5. A pose 1m from this point in the direction of the robot, and facing the labeled landmark has been selected as the goal location. The robot then navigates to this point, as can be seen in Figure 6.

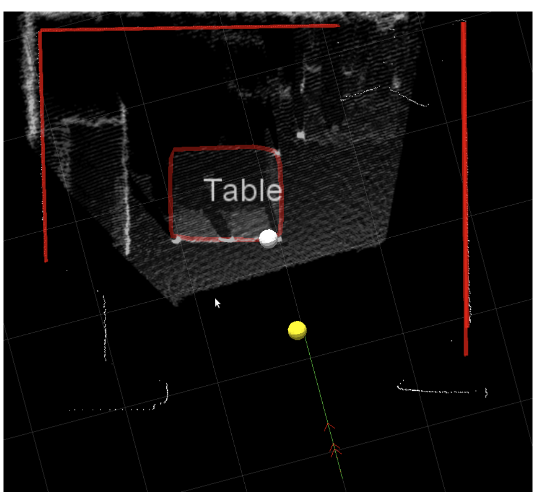

Fig. 5. A goal point corresponding to a labeled table is shown in yellow.

Navigation to a labeled region has also been tested. An example of this is shown in Figure 7, which includes a goal point that has been calculated for the label "hallway". A photo of the robot after navigating to this point is shown in Figure 8.

Fig. 6. A photo of the robot after navigating to the goal point corresponding to the table.

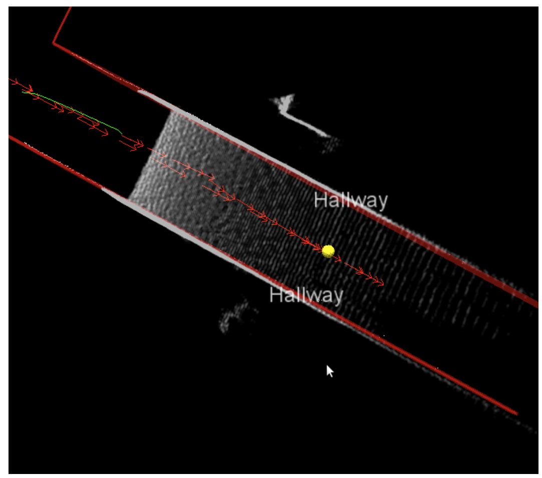

Fig. 7. A goal point corresponding to a labeled hallway is shown in yellow.

V. Discussion and Conclusion

We have proposed an approach for labeling feature based maps, and demonstrated a system implementing this approach. We described a map representation that supports labeled landmarks, such as tables, counters or shelves, as well as regions, such as rooms or hallways. We also described a user interface that allows users to annotate such maps, by entering labels on a tablet based UI, and gesturing at the landmark that should be annotated with the entered label. Landmarks throughout the environment can be labeled in a "home tour" scenario by making use of our person following system. This approach was validated by implementing and using it on our robot in our laboratory and office environment. Annotated maps were produced, and used to command the robot to go back to labeled landmarks and regions.

As future work, we will investigate the usage of the labeled maps produced by our system for service robotic tasks such as fetch and carry, object search, or other mobile manipulation tasks. Other applications might include informing active

Fig. 8. A photo of the robot after navigating to the goal point corresponding to the hallway.

visual search such as [2], or facilitating directions following tasks such as [10]. Additionally, we plan to evaluate the ease-of-use of our system with non-expert users.

VI. ACKNOWLEDGMENTS

This work was made possible through the KOR-US project, Boeing corporation and ARL MAST CTA project 104953.

REFERENCES

[1] Kai O. Arras, Óscar Martínez Mozos, and Wolfram Burgard. Using boosted features for the detection of people in 2d range data. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, Rome, Italy, 2007.

[2] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *Proc. Int. Conf. Robotics and Automation (ICRA)*, 2011.

[3] H.H. Clark and S.E. Brennan. Grounding in communication. *Perspectives on socially shared cognition*, 13(1991):127–149, 1991.

[4] F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12):1181–1204, 2006.

[5] L. Deng and X. Huang. Challenges in adopting speech recognition. *Communications of the ACM*, 47(1):69–75, 2004.

[6] M.B. Dias, TK Harris, B. Browning, EG Jones, B. Argall, M. Veloso, A. Stentz, and A. Rudnicky. Dynamically formed human-robot teams performing coordinated tasks. In *AAAI Spring Symposium ìTo Boldly Go Where No Human-Robot Team Has Gone Before*, 2006.

[7] T. Fong, F. Conti, S. Grange, and C. Baur. Novel interfaces for remote driving: gesture, haptic and pda. *SPIE Telemanipulator and Telepresence Technologies VII, Boston, MA*, 2000.

[8] H.K. Keskinpala, J.A. Adams, and K. Kawamura. Pda-based human-robotic interface. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 4, pages 3931–3936. IEEE, 2003.

[9] M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lomker, G. A. Fink, and G. Sagerer. Person tracking with a mobile robot based on multi-modal anchoring, 2002.

[10] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward understanding natural language directions. In *Proceeding of the 5th ACM/IEEE international conference on Human-robot interaction*, pages 259–266. ACM, 2010.

[11] B. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119(1-2):191–233, May 2000.

[12] M.M. Loper, N.P. Koenig, S.H. Chernova, C.V. Jones, and O.C. Jenkins. Mobile human-robot teaming with environmental tolerance. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 157–164. ACM, 2009.

[13] Oscar Martinez Mozos, Rudolph Triebel, Patric Jensfelt, Axel Rottmann, and Wolfram Burgard. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems (RAS)*, 55(5):391–402, May 2007.

[14] K. Nickel and R. Stiefelhagen. Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 140–146. ACM, 2003.

[15] K. Pathak, N. Vaskevicius, J. Poppinga, M. Pfingsthorn, S. Schwertfeger, and A. Birk. Fast 3D mapping by matching planes extracted from range sensor point-clouds. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1150–1155. IEEE, 2009.

[16] A. Pronobis, O. Martínez Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *The International Journal of Robotics Research*, 29(2-3):298, 2010.

[17] Axel Rottmann, Óscar Martínez Mozos, Cyrill Stachniss, and Wolfram Burgard. Semantic place classification of indoor environments with mobile robots using boosting. In *AAAI'05: Proceedings of the 20th national conference on Artificial intelligence*, pages 1306–1311. AAAI Press, 2005.

[18] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[19] R.B. Rusu, N. Blodow, Z.C. Marton, and M. Beetz. Close-range Scene Segmentation and Reconstruction of 3D Point Cloud Maps for Mobile Manipulation in Human Environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA*, 2009.

[20] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2422–2427. IEEE, 2004.

[21] E. A. Topp and H. I. Christensen. Tracking for following and passing persons. In *Intl Conf. on Intelligent Robotics and Systems (IROS)*, pages 70–77, Edmundton, Canada, August 2005.

[22] E. A. Topp and H. I. Christensen. Detecting region transitions for human-augmented mapping. *Robotics, IEEE Transactions on*, pages 1–5, 2010.

[23] Elin A. Topp and Henrik I. Christensen. Topological modelling for human augmented mapping. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2257–2263, Oct. 2006.

[24] A. J. B. Trevor, J. G. Rogers III, and H. I. Christensen. Planar Surface SLAM with 3D and 2D Sensors. *International Conference on Robotics and Automation*, 2012.

[25] J. Weingarten and R. Siegwart. 3D SLAM using planar segments. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3062–3067. IEEE, 2006.

[26] J. Xavier, M. Pacheco, D. Castro, A. Ruano, and U. Nunes. Fast line, arc/circle and leg detection from laser scan data in a player driver. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3930–3935. IEEE, 2005.

[27] H. Zender, P. Jensfelt, Ó.M. Mozos, G. Kruijff, and W. Burgard. An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1584. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.