To the Bookstore! Autonomous Wheelchair Navigation in an Urban Environment

Corey Montella, Timothy Perkins, John Spletzer and Michael Sands

Abstract In this paper, we demonstrate reliable navigation of a smart wheelchair system (SWS) in an urban environment. Urban environments present unique challenges for service robots. They require localization accuracy at the sidewalk level, but compromise GPS position estimates through significant multi-path effects. However, they are also rich in landmarks that can be leveraged by feature-based localization approaches. To this end, our SWS employed a map-based approach. A map of South Bethlehem was acquired using a server vehicle, synthesized a priori, and made accessible to the SWS. The map embedded not only the locations of landmarks, but also semantic data delineating 7 different landmark classes to facilitate robust data association. Landmark segmentation and tracking by the SWS was then accomplished using both 2D and 3D LIDAR systems. The resulting localization algorithm has demonstrated decimeter level positioning accuracy in a global coordinate frame. The localization package was integrated into a ROS framework with a sample based planner and control loop running at 5 Hz. For validation, the SWS repeatedly navigated autonomously between Lehigh University's Packard Laboratory and the University bookstore, a distance of approximately 1.0 km roundtrip.

1 Introduction & Motivation

In 1997, Prof. Illah Nourbakhsh of Carnegie Mellon University established *The Wheelchair Project*. Its goal was to develop an autonomous wheelchair capable of reliable navigation in both indoor and outdoor environments [14]. As an initial milestone, Nourbakhsh proposed to demonstrate autonomous navigation from Smith Hall to the University Center bookstore - a distance of approximately 500

Corey Montella
Lehigh University, 27 Memorial Drive West, Bethlehem, PA 18015, e-mail: cmontella@lehigh.edu

John Spletzer
Lehigh University, 27 Memorial Drive West, Bethlehem, PA 18015, e-mail: spletzer@lehigh.edu

meters outdoors. While it does not appear that Nourbakhsh's vision has been realized yet, we draw inspiration from his work to pose a similar challenge to ourselves: demonstrate reliable autonomous navigation of a smart wheelchair system (SWS) from Lehigh's Packard Laboratory to the University bookstore. Coincidentally, the most direct wheelchair-accessible route for our task is also ≈ 500 meters. It also required the wheelchair to navigate multiple street crossings, and to maintain a sufficiently accurate localization estimate to enable reliable navigation at the sidewalk scale.

To accomplish this objective, we were able to leverage technologies unavailable to Nourbakhsh at the time he posed the challenge. Specifically, we employed a low-cost 3D LIDAR system for both ground and obstacle detection, as well as feature tracking. We also made extensive use of Willow Garage's Robot Operating System [15]. From the development and testing that followed, we provide significant insights into what worked and what did not. By integrating these lessons learned, the SWS was capable of capable of completing our bookstore challenge without incident. Lastly, we should emphasize that while developed in support of our SWS project, these results have broad applicability to field and service robots operating in urban environments.

2 Related Work

Smart wheelchair systems (SWS) have been an active research area since the early 1980s. The spectrum of work has ranged from component level safety sensors, to assistive controllers for steering, to completely autonomous solutions. A survey of the field (as of Aug 2005) can be found in [18]. More recent projects of note include the MIT Intelligent Wheelchair Project [10], the goal of which is to develop a voice-commanded autonomous wheelchair intended for use in indoor environments. The Home, Lift, Position, and Rehabilitation (HLPR) Chair [1] developed by NIST is a special-purpose assistive mobility device to provide independent patient mobility for indoor tasks, such as moving to and placing a person on a toilet or bed. HLPR has demonstrated obstacle detection and navigation indoors with promising results. The Personal Mobility and Manipulation Appliance (PerMMA) [2] is being developed at the NSF Quality of Life Technology Center (QoLTC), with the objective of combining manipulation and mobility assistance in support of complete independence for its users. The system employs two robotic arms, and has demonstrated object manipulation tasks such as retrieving a drink from a refrigerator. Our own work to date in the smart wheelchair space includes the Automated Transport and Retrieval System (ATRS) [6]. ATRS improves automobility access for power wheelchair users by eliminating the need for an attendant to stow and retrieve the wheelchair.

In contrast to these efforts, the emphasis of our current work is navigation in unstructured, outdoor environments. Developing robust robotics solutions suitable for use outdoors is a significant challenge. Compared to indoor environments, the scale

is much larger, illumination levels vary from strong sunlight to near complete darkness, the environment is far less structured, environmental conditions can quickly and dramatically change, and simplifying assumptions such as a level ground plane cannot be used. Furthermore, operations at the sidewalk level require localization performance beyond the bounds of what traditional GPS can provide. Other research groups have studied the problem of localization in outdoor and urban environments. These include Georgiev *et al* who used a mixture of cameras, GPS, laser scanners, sonar, tilt sensors, and a database of facade models for localization [8]. They demonstrated their approach by traveling a 330 meter course and localizing with an error of approximately 1 meter. Ramos *et al* combined an EKF-SLAM approach with landmark modeling [16]. Driving an automobile over a course of 1.5 km, they were able to localize within an error of 8.6m.

In this work, we build upon our own previous results in large-scale map based localization. First, our localization approach was improved and extended to enable decimeter level accuracy. Second and more significantly, our SWS was able to employ the localization scheme for reliable *autonomous* navigation for the first time. This was validated through significant experimental results – including over 10 km of autonomous operations – culminating in the successful completion of our bookstore challenge.

3 Development Platform

3.1 Vehicle Platform

The vehicle platform used for this work is shown at Figure 1. From our experiences, three-dimensional (3D) perception is a critical enabling technology for autonomous navigation in unstructured environments. To this end, the primary exteroceptive sensor on the SWS was an IFM O3D200 3D flash LIDAR. New to the robotics arena, the IFM can measure the range to and surface reflectivity of objects in the environment. It is also relatively compact and low cost (<$1,500 US) [11]. More importantly, the IFM performs well in the range of illumination levels encountered outdoors. The trade-offs for this low price point are a relatively low resolution (48×64 pixels), narrow field of view ($30^\circ \times 40^\circ$) and limited effective range ($\approx$ 6-8 meters in our application). Despite these limitations, we could identify no sensor on the market

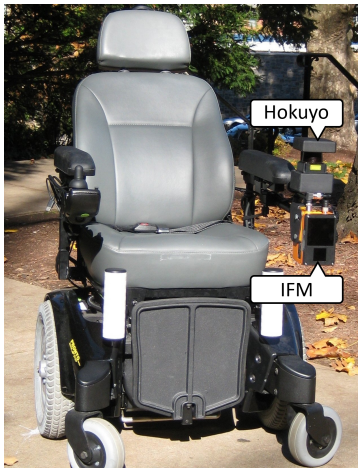

Fig. 1: The SWS vehicle integrates both 2D and 3D LIDARs for exteroceptive sensing, high-resolution encoders, and an inertial measurement unit.

that provided reliable 3D measurements outdoors at a comparable price point (note the Microsoft Kinect is not suitable for operations in bright sunlight [12]). In our application, the IFM established the ground-plane, detected obstacles, and tracked landmark features in 3D at a rate of ≈ 5 Hz.

The SWS prototype also integrates a Hokuyo UTM-30LX 2D LIDAR. The Hokuyo's larger field of view (270°), finer angular resolution (0.25°), higher update rate (40 Hz) and longer effective range (up to 20 meters) complemented the limitations of the IFM. Like the IFM, the Hokuyo was mounted as an accessory forward of the left wheelchair armrest and slightly pitched down. It was used for both obstacle detection and tracking landmark features.

Motor control for the SWS was accomplished with an on-board embedded computer. To regulate the vehicle's linear and angular velocities, it employed a software-based PID that controlled the individual wheel velocities using feedback from high resolution quadrature encoders (4,096 CPR). A Microstrain 3DM-GX1 inertial measurement unit mounted to the SWS frame provided gyro corrections for improved odometry performance. All other processing was done by a laptop computer with a 1.6 GHz Intel 720QM processor.

3.2 Software Architecture

The system software architecture was based upon the ROS paradigm of interconnected nodes that communicate via messages [15]. Each node subscribed to topics published by other nodes, and published its own messages as it processes data. Nodes also listened to a transform tree, which allowed messages to transform to and from any frame. When processing LIDAR data, we leveraged ROS' Point Cloud Library (PCL) which provided basic algorithms like RANSAC segmentation, pass-through filtering, and nearest neighbor cluster extraction [20]. The block diagram in Figure 2 shows the basic structure of the system architecture. These are discussed in greater detail in the following sections.

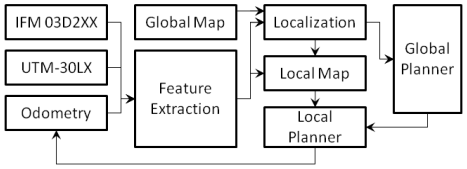

Fig. 2: Major functional blocks of the SWS software architecture.

4 Generating the Global Map

A primary motivation for the map-based localization approach was that by leveraging such maps, robots with lower cost sensor suites gain some of the benefits obtained from the higher fidelity sensors of the mapping platform, but without the

cost. This necessitated an *a priori* map construction phase for both localizing landmarks, as well as identifying wheelchair-accessible routes within the global map. These two phases are described below.

4.1 The Landmark Map

To obtain the desired sub-meter localization accuracy in the absence of reliable GPS, we relied upon a map-based localization approach. This necessitates the availability of an accurate landmark map. To this end, we leveraged our previous results in large-scale mapping [7]. In this paradigm, three dimensional map data were acquired by manually driving an automobile (Figure 3) instrumented with a high precision GPS/INS system, in conjunction with light detection and ranging (LIDAR) systems. LIDAR range measurements were subsequently registered to a global (WGS 84) coordinate frame. The resulting map data were then synthesized *a priori* to identify robust, salient features for use as landmarks in localization.

As in [7], the landmarks that were used in this work were limited to "pole-like" features (*e.g.*, lamp posts, trees, parking meters, street signs, *etc.*) that are prevalent in urban landscapes. Each landmark was characterized by 5 parameters: 1) *x*-*y* position (WGS 84), 2) positional uncertainty (covariance) estimate, 3) radius estimate, 4) radius estimate uncertainty, and 5) reflectivity. The radius was used to classify the landmarks into one of seven classes, with examples shown at Figure 4. This classification was used to protect against incorrect landmark data association, and is described in more detail in Section 6.2.

Fig. 3: The mapping vehicle integrated an OXTS RT-3050 for 6DoF vehicle pose estimation, and a pair of roof mounted Sick LMS291-S14 LIDARs (circled red) to capture relative range measurements.

We should also note that while the original landmark map of South Bethlehem had been automatically synthesized as outlined in [7], it was augmented for this work "by hand" with additional landmarks. This was necessitated due to the recent razing and reconstruction of a one square block region along our route, and the unavailability of the mapping vehicle during this time frame. The coordinates of larger features (*e.g.*, lamp posts, trees, *etc.*) were identified from satellite imagery, while smaller features (*e.g.*, parking meters and street signs) were mapped by the SWS itself, and subsequently refined during a SLAM phase to improve local consistency. After this refinement phase, their locations and positional covariances were considered fixed within the landmark map.

Fig. 4: Instances of the 7 landmark classes. Landmarks were classified and tracked based upon geometry and reflectivity. Detected landmarks could be reliably associated with their respective classes which prevented data association errors in our experiments.

4.2 The Route Network

The SWS relies upon a route network that exists in the world frame, and serves as the global map for wheelchair accessible paths. The route network was constructed by manually driving the SWS along the desired navigation path while running the localization algorithm described in Section 5. Waypoints were then obtained by sampling the SWS pose for every ≈ 4 meters traveled. In addition to WGS 84 position information, waypoints were also augmented with semantic information to include labels (*e.g.*, "LU Bookstore"), speed limits, and stop points which the robot must obey. Stop points were placed at areas such as cross walks which currently require human supervision to safely traverse. In our current implementation, when the SWS reaches a designated stop point it will pause until receiving a resume command from the operator.

For global path planning, the route network was represented as a graph $G(V,E)$. Waypoints in the route network corresponded to vertices $v_i \in V$ of G, and the edge set $E \subseteq G$ corresponded to path segments where each $e_{ij} \in E$ connected a pair of waypoints (v_i, v_j). Edges were weighted based upon the expected traversal time as estimated using the edge length and associated waypoints' speed limits. The optimal path to a given destination was then specified via a waypoint sequence as determined from Dijkstra's algorithm.

5 Map-based Localization

Localization of the SWS was based upon a modified version of the FastSLAM 2.0 algorithm [13]. Although FastSLAM is a localization and mapping algorithm, for this work no mapping was conducted during navigation, as the fixed landmark map was provided a priori. The algorithm was also adapted to accommodate multiple observations per control, as the control loop (5 Hz) and LIDARs (5 Hz and 40 Hz) operated asynchronously and at different update rates. Each particle was of the form

$$Y_t^{[k]} = \langle x_t^{[k]}, \langle \mu_1^{[k]}, \Sigma_1^{[k]}, s_1^{[k]} \rangle, ..., \langle \mu_N^{[k]}, \Sigma_N^{[k]}, s_N^{[k]} \rangle \rangle \quad (1)$$

where $x_t^{[k]}$ was the pose of the k^{th} particle at time t. Each of the N landmarks was parameterized by a mean position $\mu^{[k]}$ and covariance estimate $\Sigma^{[k]}$ which remained fixed. Furthermore, each landmark carried additional semantic data used for data association and tracking. In our case, this was the radius of the landmark and reflectivity (in the tracking case). For our tests we used a fixed set of 60 particles. While fewer could be used when the robot was properly localized, we found a large set of particles (> 30) was important to properly initialze the robot.

For the prediction phase of the filter, we sampled from a probabilistic motion model for a differential drive robot where the robot control inputs (v, ω) were corrupted with additive Gaussian noise [19]. Four noise parameters were used, two to indicate control noise for a traslational movement ($a_1 = .05$ and $a_2 = .01$), and two to indicate control noise for a rotational movement ($a_3 = .001$ and $a_4 = .1$). We determined the values for these parameters empirically. Further, we found that with sparsity of landmarks a diverse particle set improves localization performance, thus exaggerated noise in the motion model was preferred.

To perform data association with the landmarks, we used maximum likelihood correspondence (MLC) [19]. Our MLC implementation compared each observation to every landmark in the local (cost) map, and computed a weight for each association. The weight was approximated by a Gaussian with mean $(z_t - \hat{z_j})$, where z_t is the observation at time t and $\hat{z_j}$ is the predicted observation of landmark j, and covariance $Q_j = H_j \Sigma_j H_j^T + Q_t$, where H_j is the 3x3 pose Jacobian taken with respect to map features, Σ_j is the covariance of the j^{th} landmark, and Q_t is the linearized vehicle measurement noise. The observation and predicted observation are of the form $z_t = [\rho_t, \phi_t, s_t]^T$ where ρ, ϕ, and s are respectively the range, bearing, and radius of a feature. The final weight is then approximated by

$$w_j = |2\pi Q_j|^{-\frac{1}{2}} exp\{-\frac{1}{2}(z_t - \hat{z_j})^T Q_j^{-1}(z_t - \hat{z_j})\} \tag{2}$$

The landmark with the maximum weighted association w^* was then used for localization as long as it exceeded a minimum threshold.

To assess the performance of our localization module, we measured the real world distance from 22 landmarks and compared this agains the localized robot pose. The robot was manually driven through the course and at 22 points we measured the distance from the robot to nearest landmark. This same distance was calulated through analyzing recorded data. We repeated this for three different scenarios: perfect observation, observation of only 75% landmarks, and observation of only 50% landmarks. This was done to simulate cases where landmarks may be occluded by dynamic obstacles (*i.e.*, pedestrians). The results of this experiment are summarized in Table 1. While far from exhaustive, these results indicate that while localization accuracy does drop as landmarks are occluded, average 1D accuracy remained sub-decimeter when even half the landmarks were not observed. More significantly, the localization filter did not diverge.

	100% Landmarks	75% Landmarks	50% Landmarks
Average Error (cm)	5.15	5.97	7.18
Standard Dev (cm)	4.81	5.20	6.48
Maximum Error (cm)	15.71	18.28	23.20
Minimum Error (cm)	0.16	0.38	0.31

Table 1: Sub decimeter 1D localization accuracy is achieved even when half the landmarks are removed from the map. In each scenario the localized robot pose did not diverge.

6 Perception

6.1 Ground/Obstacle Detection

Obstacles were detected by both the IFM and Hokuyo LIDARs. The IFM accounted for close range (within 6m) and low lying obstacles. The first step to obstacle detection was reliable segmentation of the ground plane. This was accomplished using RANSAC to robustly fit the IFM point cloud measurements [5]. Once an accurate ground plane was established, all inlier points were removed from the raw scan. The remaining scan was then filtered to remove outliers, resulting in a cloud of obstacle points.

The Hokuyo was able to detect obstacles that were further away (up to 20m) and above the ground plane. Obstacles detected by either the IFM or Hokuyo LIDARs were broadcast for integration into the local map. This is discussed in more detail in Section 7.1.

6.2 Landmark Feature Segmentation

Both the IFM and Hokuyo LIDARs were employed for landmark segmentation. However, since 2D and 3D landmark representations are dramatically different, the segmentation approaches also differ.

For the 2D case (Hokuyo LIDAR), each scan was first registered to the fixed global frame to account for any wheelchair motion between scans. Each scan was then decomposed into connected components with a maximum intra-cluster spacing of 45 cm, and a minimum cluster size of 4 points. Then, a 2D circle was fitted to each connected component using RANSAC [5]. Circles with a fitted radius greater than 40 cm were discarded from consideration, as these features were larger than any in our landmark map. Finally, any cluster that did not fit the circle model well (less than 90% inliers) was discarded. The remaining components were then tracked as potential landmarks. We analyzed each connected component to determine its location in the global frame, its range and bearing to the robot, the dimensions of its bounding box, its radius, its reflectivity, and the number of observations of that feature.

To discriminate between static features (*e.g.*, landmarks, stationary pedestrians, *etc.*) and dynamic features (*e.g.*, walking pedestrians, moving cars), each new feature was compared against a list of features detected in previous scans. New features that were within the 45 cm intra-cluster threshold and with intrinsic parameters (bounding box, radius, and reflectivity) within 10% of a previously detected feature will increment the observation count of that previous feature, as well as update all intrinsic parameters. The range and bearing to a feature was only published as an observation for the localization engine when that feature was observed at least twice.

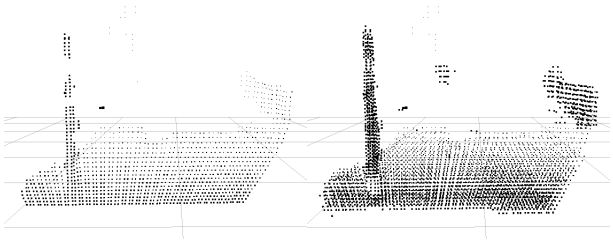

Fig. 5: Scan fidelity was improved by creating composite images from multiple scans. A single scan of a pole (left) shows three distinct clusters. Only the bottom will be tracked. The pole in the combined scan (right) is contiguous, and will be tracked more reliably due to its higher height/width ratio.

The 3D case was similar, except given the extra dimension we could infer more about the intrinsic parameters of objects. Since the LIDAR scans were sparse, we concatenated the last 5 scans (again registered to the global frame) to create a richer point cloud for feature extraction as shown in Figure 5. We ran this input cloud through a series of filters to remove noise. The ground plane was then removed, and the remaining cloud was further processed to segment features. As in the 2D case, the point cloud was broken into connected components of nearest neighbor clusters with a maximum intra-cluster spacing of 7 cm. A cylindrical model was then fitted to each connected component, again using RANSAC, which allowed us to estimate the component's position, orientation, and radius. We necessitated that any feature have a height/width ratio of at least 2, and the lowest point of the feature should be at most 5 cm from the ground plane. Again, the range and bearing to a feature was only published as an observation when that feature was observed at least twice.

7 Planning & Control

7.1 Generating the Cost Map

As the wheelchair traversed its environment, it maintained a local cost map represented as a 2D occupancy grid [4]. The cost map was 20 m $\times$ 20 m in size centered on the robot base, and followed the vehicle in a rolling window fashion. Cell resolu-

tion was 5 cm. Each cell was assigned a cost based on three factors: distance to path (C_p), distance to goal (C_g), and occlusion cost (C_o). Cells containing the global path were marked with zero path cost. For all other cells, C_p was proportional to the Euclidean distance from each cell to the closest path cell. A similar process was used to calculate the C_g. Note however that the goal was not the final goal destination, but the last global path point within the dimensions of the rolling window. Any cell that contained an obstacle (as determined by the IFM or Hokuyo) was marked as occupied and given maximum C_o which would prohibit traversal. The cost of these cells was inflated to a radius of 80cm, or the circumsribed radius of the SWS footprint, as described in [3]. Occupied cells were only cleared when a ray from the laser can be traced through the occupied cell. The cost map served as input to the local planner for trajectory optimization. A sample costmap is shown in Figure 6.

7.2 The Local Planner

The SWS employed trajectory rollout for local planning [9]. This is a sample based approach on the input space of the linear and angular control velocities (v, ω). A sample trajectory $T = \{\mathbf{x_0}, v_1, \omega_1, ..., v_k, \omega_k\}$ was specified by the current robot pose $\mathbf{x_0}$ and a sequence of k velocity inputs where k denotes the number of timesteps in the control horizon. The velocities were then integrated forward in time yielding a projected path over the chosen time horizon. An advantage of sampling the control velocities is that we ensure each trajectory is feasible in terms of the wheelchair kinematics. For our implementation, the discretization of v and ω was .06 m/s and .05 rad/s, respectively, and the control horizon $k = 10$ time steps. Each trajectory T_i was then evaluated against the cost map M described in Section 7.1 using the cost function $C(T_i, M) = k_o C_o + k_p C_p + k_g C_g$. The optimal trajectory $T^* = \arg\min C(T, M)$ was then selected, and the associated velocity command $(v_1^*, \omega_1^*) \in T^*$ was issued to the wheelchair controller.

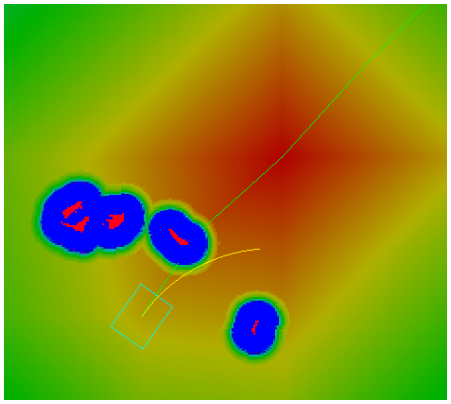

Fig. 6: Navigation visualization. The blue square is the robot, the green line is the desired path, the yellow line is the lowest cost trajectory. The bright red cells are obstacles of maximum cost. Obstacle cells are inflated with a high cost region in blue.

One further refinement was made to model actuator latency. The local planner maintained a queue of the two most recent velocity inputs to the motor controller. These were then used in a feed-forward fashion to deterministically move the SWS prior to generating trajectories for evaluation.

8 Experimental Results

Fig. 7: Satellite view of the route from Packard Laboratory to the Lehigh bookstore. Stop points are shown as red octagons, and landmarks as red "x" marks. The round-trip distance is 980 meters. The four numbered points correspond to locations of interest, as highlighted in Figure 8

Fig. 8: Close-ups of points of interest along the route from Packard Laboratory to the LU Bookstore. These include (1) a crosswalk on Packard Ave., (2) street crossing at Asa Packer Drive with curb cutouts, (3) narrow sidewalk Morton St. with significant obstacles, and (4) the route destination.

Figure 7 provides a satellite view of our bookstore challenge. The green line segments denote the route from Packard Laboratory (right side) to the bookstore. Note that a more direct route was not used, as stairs made it inaccessible for wheelchair users. Red "octagons" denote stop points and are located at the two street crossings as well as the entrance to a parking garage. As noted previously, when the SWS reaches a stop point, it will pause until manually resumed by the operator (*i.e.*, the user touches the space bar). In a clean run, these and specifying the goal location would be the only inputs provided by the user. The red "x" marks denote the lo-

cations of landmarks along the route. A maximum of 92 could be observed by the SWS along the route.

The numbered locations in Figure 7 correspond to the sub-figures in Figure 8 to provide a close-ups of points of interest. These include: 1) the cross-walk on Packard Avenue, 2) the street crossing on Asa Packer Drive, 2) he sidewalk on Morton Avenue, which is only wide enough for single direction traffic, and 4) the Lehigh University (LU) bookstore. A successful trip would bring the SWS immediately in front of the bookstore doors.

Over the course of a one month test period, a total of 11 round-trip missions to/from the bookstore were conducted, for a total distance over 10 km. The nominal SWS speeds were 0.9 m/s (3.1 km/hr) for runs 1-4, 1.0 m/s (3.5 km/hr) for runs 5-8, and 1.2 m/s (4.3 km/hr) for runs 9-11. Of the 11 trips, 3 required user intervention.

The first intervention occurred during the low speed trials, and was attributed to excessive inflation of obstacles in the cost map. This lead the SWS to believe incorrectly that the sidewalk path was blocked. In this case, the SWS attempted to circumvent the perceived obstacle by traveling along the grass to the right. Although this behavior would have been safe, it was obviously incorrect and the run was aborted. A subsequent adjustment to the cost map parameters was made, and this failure mode was not seen subsequently. Success in subsequent trials lead us to increase the SWS speed to 1.0 m/s, and it was this higher speed that exhibited the second failure mode.

At 1.0 m/s, two of the first three trials ended with operator interventions as a result of the SWS leaving the sidewalk path. In one case, a drive wheel migrated onto the grass shoulder and the second time into a mulch bed. Upon review of the log files, we determined that although the SWS perception and planning subsystems were operating properly and sending correct velocity inputs to the motor controller, these were not being actuated in a timely fashion. The cause of failure was identified as the motor controller. The motor controller PID gains had originally been tuned to minimize the steady-state error in the wheel velocities. However, this came at the expense of rise time. As a result, the PID could not achieve the velocity setpoints in a single 200 ms I/O cycle and short duration velocity inputs were lost. This deficiency had been masked by the overall system latency, and as a result the local planner could not compensate for the lost actuator inputs. This shortcoming was corrected by re-tuning the motor controller PID using a minimum rise time criterion which was more suitable for real-time control. This allowed setpoints to be approached in a single 200 ms control cycle. An additional benefit from this change was a reduction in the overall actuation latency from approximately 600 ms to 370 ms as estimated by cross-correlating the velocity I/O response.

After these modifications, an additional trial with the SWS was conducted at the 1.0 m/s velocity. Wheelchair response was noticeably improved, and the trial was completed successfully. As a result of this success, the velocity was further increased to 1.2 m/s, and three trials were completed at this speed without incident. This corresponded to a total distance of over 3.8 km with the new PID tuning, and the SWS behaved predictably at all times. A video of one of these trials can be

viewed at http://youtu.be/FUgHMReg4xM. Screenshots from this run for the same areas of interest discussed above are at Figure 9.

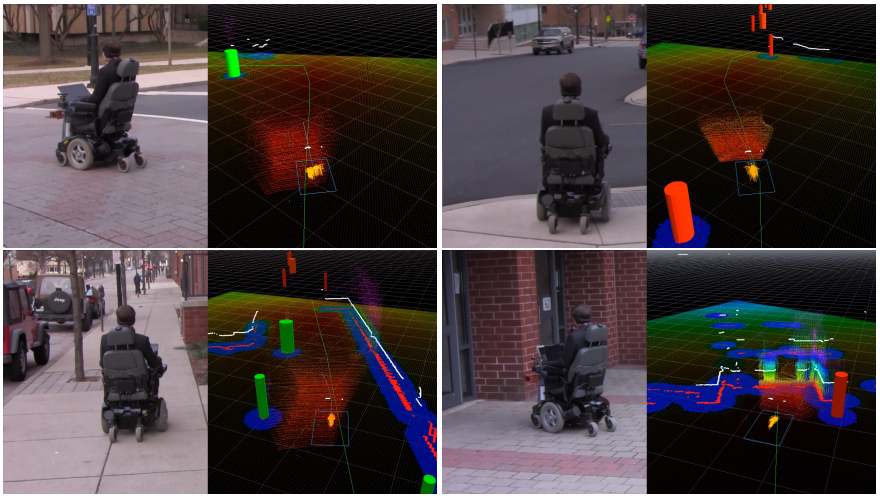

Fig. 9: Screen captures of a navigation run. Video capture is on the left, visualization of data is on the right.

9 Conclusions

In this work, we successfully completed our bookstore challenge to demonstrate autonomous navigation of a smart wheelchair system in an urban environment. This was realized by combining 2D and 3D LIDAR sensor systems with a global map to obtain decimeter level localization accuracy. Open-source software (ROS/PCL) was also leveraged to facilitate platform development.

While we are satisfied with our progress to date, significant work in many aspects of autonomous navigation remains before real-world systems will be available. In the immediate future, we intend to investigate higher speed limits for the SWS, and to grow the global map to investigate longer distance/duration operations. We have no doubt that additional failure modes will manifest as we expand the wheelchair's operational envelope.

A primary future emphasis will be navigation in crowds. This will require SWS localization to be robust to occlusion for intermittent periods of time, and likely require the introduction of additional landmark classes into the global map. It will also necessitate reliable people detection and tracking so that appropriate interaction models can be made. We have obtained preliminary results in this area [17], but effectiveness has been hindered by the constrained field-of-view ($30^\circ \times 40^\circ$) of the IFM. In truth, a significantly wider field-of-view would be greatly beneficial for all aspects of SWS navigation. Still, we will continue to make progress, which will inevitably be aided by the future maturation of sensor technologies.

References

1. Bostelman R., Albus J., "Sensor Experiments to Facilitate Robot Use in Assistive Environemnts," Proceedings of the 2008 International Conference on PErvasive Technologies Related to Assistive Environments, Athens, Greece (2008)
2. CMU Quality of Life Technology Center, "Personal Mobility & Manipulation Appliance (PerMMA)," http://www.cmu.edu/qolt/Research/projects/current-projects/permma.html Cited 13 Jan 2012
3. Eitan Marder-Eppstein, "costmap_2d," http://ros.org/wiki/costmap_2d Cited 13 Jan 2012
4. Elfes, A., "Using occupancy grids for mobile robot perception and navigation," IEEE Computer Magazine, **22**, 56-57 (1989)
5. Fischler M., Bolles R., "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Communications of the ACM (1981)
6. Gao C., Hoffman I., Miller T., Panzarella T., Spletzer J., "Autonomous Docking of a Smart Wheelchair for the Automated Transport and Retrieval System (ATRS)," Journal of Field Robotics, **25**, 203-222 (2008)
7. Gao C., Sands M., Spletzer J., "Towards Autonomous Wheelchair Systems in Urban Environments," Proceedings of the 2009 International Conference on Field and Service Robotics, Cambridge, Massachusetts (2009)
8. Georgiev, A. and Allen, P.K., "Localization methods for a mobile robot in urban environments," IEEE Transactions on Robotics, **20-5**, 851-864 (2004)
9. Gerkey B. P., Konolige K., "Planning and Control in Unstructured Terrain," http://pub1.willowgarage.com/apubdb_html/files_upload/8.pdf Cited 6 Jan 2012
10. Hemachandra S., Kollar T., Roy N., Teller S., "Following and Interpreting Narrated Guided Tours," Proceedings of the 2011 International Conference on Robotics and Automation, Shanhai, China (2011)
11. IFM Effector, "3D Sensor Wins Design News Magazine's 2009 Best Product of the Year - Sensors and Vision Category." http://www.ifm.com/ifmus/web/prod_tip_3d.htm Cited 6 Jan 2012
12. Microsoft, "Xbox Support - Lighting," http://support.xbox.com/en-us/pages/kinect/more-topics/lighting.aspx Cited 6 Jan 2012
13. Montemerlo M., Thrun S., Koller D., Wegbreit B, "FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges," Proceedings of the 2003 International Joint Conferences on Artificial Intelligence, Acapulco, Mexico (2003)
14. Nourbakhsh, I.R., "The Wheelchair Project," http://www.cs.cmu.edu/ illah/wheelchair.html Cited 21 Dec 2011
15. Quigley M., Conley K., Gerkey B.P., Faust J., Tully F., Leibs J., Wheeler R., Ng A.Y., "ROS: an open-source Robot Operating System," ICRA Workshop on Open Source Software (2009)
16. Ramos, F., Nieto, J. and Durrant-Whyte, H., "Recognising and Modelling Landmarks to Close Loops in Outdoor SLAM," Proceedings of the 2007 International Conference on Robotics and Automation, Roma, Italy (2007)
17. Savtchenko, C. and Spletzer, J., "Sidewalk-level People Tracking with a Low-cost 3D LIDAR System," Lehigh University Technical Report LU-CSE-11-003 (2011)
18. Simpson R.: Smart wheelchairs: A literature review. Journal of Rehabilitation Research and Development, **42**, 423-436 (2005)
19. Thrun S., Burgard W., Fox D., "Probabilistic Robotics," The MIT Press, Cambrige, Massachusetts (2006)
20. Willow Garage, "Point Cloud Library," http://pointclouds.org Cited 13 Jan 2012