Towards a Safe, Low-Cost, Intelligent Wheelchair

Aniket Murarka Shilpa Gulati Patrick Beeson Benjamin Kuipers

Abstract— **Unlike most other robots, autonomous personal transports must be designed with a passenger user in mind. This paper examines the integration of three necessary technologies for a robotic transport—in particular, a robotic wheelchair. First, local motion to a nearby goal pose needs to be safe and comfortable for the human passenger. Second, 3D overhangs, drop-offs, steep inclines, and stairs (in addition to pedestrians and walls) need to be accurately modeled and avoided, while curb cuts, drivable ramps, and flat ground should be seen as traversable. Third, the spatial representation of the robot should facilitate infrequent requests for human directions and allow "natural" directional commands. Furthermore, the sensorimotor system that facilitates spatial reasoning, planning, and motion needs to be cost efficient. As a result, our goal is to create a system that ultimately uses inexpensive wheel encoders and off-the-shelf stereo cameras. In this paper, we overview the three technologies listed above. We then discuss the successes and the current failures of the integration task, both of which motivate future work.**

I. Introduction

The Intelligent Wheelchair is designed to serve as a mobility aid for a human driver. It is also an autonomous robotic agent that learns the spatial structure of its environment from its own experience and is able to act autonomously in pursuit of goals set by the human. The robot acts as a *chauffeur* for the human. The current physical instantiation of the Intelligent Wheelchair is shown in Figure 1.

The Intelligent Wheelchair's cognitive architecture uses the *Hybrid Spatial Semantic Hierarchy (HSSH)* [1], [2], which integrates four different representations for knowledge of space. By using multiple spatial knowledge representations, the wheelchair supports different modes of interaction and different levels of autonomy. In this paper, we deal with the inference and control at the lowest level of the HSSH hierarchy, which in turn affects the higher levels. Section II briefly overviews the HSSH.

Previous HSSH implementations used planar lidar sensors for reliable detection of obstacles at a fixed height from the ground plane. This allowed straight-forward SLAM (simultaneous localization and mapping) inference using 2D metrical maps in the HSSH Local Metrical level. For the Intelligent Wheelchair, we wish to overcome the need for expensive and/or bulky sensors like lidar.

This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by grants from the Texas Advanced Research Program (3658-0170-2007), and from the National Science Foundation (IIS-0413257, IIS-0713150, and IIS-0750011).

A. Murarka is with the Dept. of Computer Sciences, University of Texas at Austin

S. Gulati is with the Dept. of Mechanical Engineering, UT Austin

P. Beeson is with TRACLabs, Inc.

B. Kuipers is with the Dept. of Electrical Engineering and Computer Science, University of Michigan

Fig. 1. The current Intelligent Wheelchair platform. A human "driver" can either use the joystick or GUI interfaces via a laptop. There is a stereo camera on a pan-tilt unit, one horizontal lidar, and one vertical lidar (currently unused). The horizontal lidar is used for efficient SLAM; however, future platform configurations should eliminate expensive lidar sensors and utilize visual SLAM [3] along with the visual 3D modeling discussed below.

Additionally, we wish for our robot to handle common non-planar situations, including drop-offs, inclines, and overhangs; thus motion planning algorithms need good models of the 3D local surround. This work demonstrates how the HSSH Local Metrical representation can be created using off-the-shelf stereo cameras. The representation facilitates safe navigation in non-planar environments. Section III overviews the process of creating a 3D hybrid model of *small-scale space* (space immediately surrounding the robot) and discusses how this is transformed into the 2D *Local Perceptual Map* of the HSSH Local Metrical level.

The grid-based Local Perceptual Map (LPM) is useful for efficient planning around obstacles, while avoiding drop-offs and overhangs. Previous HSSH implementations dealt with control at this lowest level in a quite ad hoc fashion, simply following a piecewise linear plan at a constant velocity; however, for a passenger transport, comfortable yet safe trajectories must be created. Section IV demonstrates fast planning of trajectories that result in motion that is comfortable for the human passenger.

Section V discusses the integration successes of the 3D vision-based model and the comfortable trajectory algorithm into the HSSH. We show examples of environments that the robot was unable to navigate with lidars, but can now successfully navigate using the vision-based model. However, like all integration challenges, there are some failures and/or unexpected degradations compared to previous HSSH implementations based on planar lidar maps and linear plans. These problems motivate interesting short-term and long-term future work, which is outlined in Section VI.

II. HSSH Overview

The Spatial Semantic Hierarchy (SSH) [4] is a spatial representation framework inspired by the multiple layers of knowledge that humans utilize in navigating large-scale spaces. This framework is extended to the Hybrid Spatial

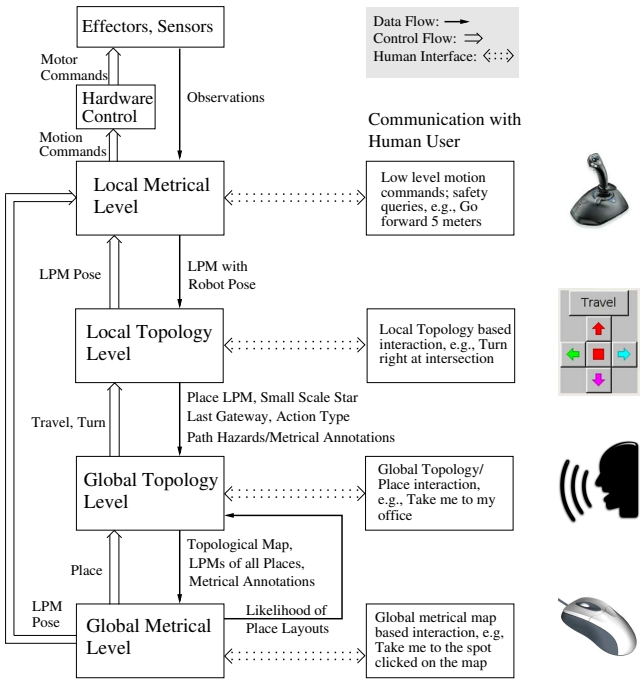


Fig. 2. The cognitive architecture of the Intelligent Wheelchair is based on the HSSH. The rightmost column shows the interface modes we are currently investigating for each representational level.

Semantic Hierarchy (HSSH) by incorporating knowledge of local (small-scale) perceptual space. The Intelligent Wheelchair's cognitive architecture, which is the HSSH, is illustrated in Figure 2.

The HSSH has four major levels of representation. At the *Local Metrical level*, the agent builds and localizes itself in the Local Perceptual Map (LPM), a metrically accurate map of the local space within a bounded sensory horizon. The LPM is used for local motion planning and hazard avoidance. At the *Local Topology level*, the agent identifies discrete places (e.g., corridor intersections, rooms, etc.) in the LPM discretization of the small-scale environment, and qualitatively describes the configuration of the paths through the place—its local decision structure. At the *Global Topology level*, the agent resolves large-scale structural ambiguities (e.g., loop closing) and determines how the environment is best described as a graph of places, paths, and regions. The *Global Metrical level* specifies the layout of places, paths, and obstacles within a single global frame of reference. It can be built on efficiently using the loop-closing constraints provided by the topological map [2]. It is useful when available, but is often unnecessary for large-scale navigation.

The human-robot interaction levels for the Intelligent Wheelchair correspond with the distinct representations in the Hybrid Spatial Semantic Hierarchy. The higher levels of interaction require more intelligence on the part of the wheelchair, but they also require less effort for communication and supervision by the human driver. In order to maximize human autonomy, the driver can shift freely between the different levels at any time.

A. Local Metrical Modeling

Humans have relatively reliable metrical models of their nearby, local surroundings. Likewise, the Intelligent Wheelchair builds and maintains a fixed-size Local Perceptual Map (LPM) that is centered on the wheelchair and follows its motion while describing the wheelchair's small-scale surroundings. The frame of reference of the LPM may drift with respect to the global frame, but this is resolved in the Global Topological and Global Metrical levels.

Regions in the LPM can be classified as free space, obstacles, or unknown space. Obstacles in the LPM can be further classified as static or non-static, making it possible to identify dynamic hazards such as pedestrians and structures such as doors (that change the apparent topology of places). Previous versions of the HSSH represented the LPM as a fixed-size occupancy grid map built using lidar sensors using existing methods for simultaneous localization and mapping (SLAM) [5]. The current implementation of the HSSH uses vision to build the LPM as discussed in Section III.

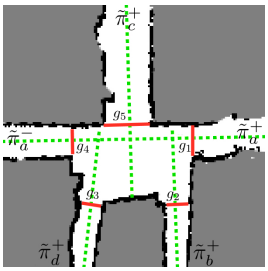
B. Local Place Topology

Humans generate symbolic descriptions of the navigational affordances of the local space, and therefore understand its qualitative decision structure. An Intelligent Wheelchair should understand terms the human driver finds useful and comfortable, including navigation commands that presuppose knowledge of the local decision structure, such as "Turn right" or "Take the second left". Fortunately, these terms correspond well with the HSSH Local Topology level.

As the Intelligent Wheelchair moves through the environment, it maintains the LPM as an accurate metrical model of local small-scale space. From the LPM it describes the local topology of nearby space in terms of *local paths* and *gateways* (see Figure 3(a)). Local paths are the navigation affordances provided by the motion control laws that support travel across the boundary of the LPM. Gateways are divisions across those local paths, separating the core of the local region from its boundaries. Details on the current robust gateway algorithm are provided in previous work [1]. For the experiments in this paper, the wheelchair creates a 140x140 cell LPM, which means gateways can be computed at $\sim$8 Hz.

Local paths in small-scale space correspond to the locally visible portions of topological paths in large-scale space. When the LPM contains exactly two gateways, and they align sufficiently well to lie on a single, unique local path, the agent describes itself as *between places*, traveling along a path. Any other configuration of gateways and local paths requires a navigation decision, so the local neighborhood defines a topological *place*.

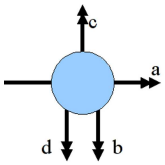
The local place topology is described as a circular order of directed local paths and gateways, which translates directly to the large-scale space description of a place as a node in a graph, connected to paths (see Figure 3(c)). In previous work, we discussed the formal mapping between small-scale and large-scale ontologies [2]. In large-scale space (where most route planning takes place), a command such as "Turn left" selects an outgoing directed path, given the incoming one.



(a)

$$
\begin{array}{lcl}
(\langle \tilde{\pi}_a^+, 1\rangle & \leftrightarrow & \langle g_4, in\rangle, \langle g_1, out\rangle) \\
(\langle \tilde{\pi}_b^+, 1\rangle & \leftrightarrow & \langle g_2, out\rangle) \\
(\langle \tilde{\pi}_c^-, 0\rangle & \leftrightarrow & \langle g_5, in\rangle) \\
(\langle \tilde{\pi}_d^+, 1\rangle & \leftrightarrow & \langle g_3, out\rangle) \\
(\langle \tilde{\pi}_a^-, 1\rangle & \leftrightarrow & \langle g_1, in\rangle, \langle g_4, out\rangle) \\
(\langle \tilde{\pi}_d^-, 0\rangle & \leftrightarrow & \langle g_3, in\rangle) \\
(\langle \tilde{\pi}_c^+, 1\rangle & \leftrightarrow & \langle g_5, out\rangle) \\
(\langle \tilde{\pi}_b^-, 0\rangle & \leftrightarrow & \langle g_2, in\rangle))
\end{array}
$$

(b)



(c)

Fig. 3. (a) From a snapshot of the current LPM, local paths $\tilde{\pi}$ and gateways g are found and aligned. (b) Based on gateway alignment, a symbolic circular ordering of the local space is created. (c) In large scale space, places are abstracted to 0D points with 1D circular orderings of paths.

In small-scale space (where motion control actually takes place), the same command translates to a specific gateway. This, in turn, gets translated into a goal location in the LPM frame of reference, which seeds planning and local motion.

By instructing the wheelchair at the Local Topology level, the human driver is delegating the autonomy for hazard-avoiding travel between places, for recognizing the decision structures of places, and for selecting the intended option at that place. Thus, at the Local Topology interaction level, both the driver and the Intelligent Wheelchair represent the space as a graph of decision points. In the current implementation, the available commands are Forward, Right, Left, and Turn Around as indicated by the GUI window shown in Figure 2.

C. Global Topological and Metrical Maps

People tend to solve way-finding problems primarily based on their topological knowledge of the environment. The HSSH builds a global topological map from the agent's travel experience, expressed as a sequence of places with local topologies, linked by travel actions. People also often find it helpful to use metrically accurate graphical maps. A global metrical map in a single frame of reference can be built efficiently using the global topological layout and stored metrical displacements from the Local Metrical Level [2]. In this paper, we ignore the Global Topology and Global Metrical Levels, as they remain unchanged given quality abstractions of local topology from the LPM.

III. Modeling 3D Surfaces

This section gives an overview of the stereo vision-based algorithm for 3D modeling of the robot's surround. The algorithm and its quantitative evaluation are detailed in other work [6]. The goal of this algorithm is to create a drop-in replacement for the traditional lidar-based occupancy grid Local Perceptual Map (LPM). If successful, the vision-based Local Perceptual Map should facilitate safe control at the Local Metrical level as well as support inference and control at the Local Topology level (and the global levels) of the HSSH—even in non-planar environments. Section V details progress in integrating the vision-only LPMs with the existing HSSH codebase.

As the robot explores its local surroundings, it receives a constant stream of stereo images. Each time the robot gets a new stereo image pair, it processes the images to update its current knowledge of the world. The following steps are involved in processing each frame in order to produce an LPM at the Local Metrical level of the HSSH.

1) A disparity map relative to the left image is computed using the camera's built-in correlation stereo method [7] (Figures 4(a) & 4(b)). The range readings obtained are transformed into the LPM frame of reference using localization.

2) A 3D model consisting of a 3D grid (Figure 4(c)) and a 3D point cloud (Figure 4(d)), is updated with the range readings obtained above using an occupancy grid algorithm. The 3D point cloud is obtained by maintaining a list of the range points that fall in each occupancy grid voxel.

3) Planes are fit to potentially traversable ground regions in the 3D model using a novel plane fitting algorithm consisting of two steps. First, ground regions are found by segmenting the 3D grid based on the heights of voxels columns—Figure 4(e) shows the segments identified. Second, planes are fit using linear least squares to points corresponding to the segments (Figures 4(f) & 4(g)).

4) Finally, the segments and planes are analyzed for safety to yield an annotated 2D grid map called the *local safety map* (Figure 4(h)) that tells the robot which regions are known to be safe (or unsafe) at the current time. Each cell in the map is annotated with one of four labels: *Level:* implying the region in the cell is level and free of obstacles; *Inclined:* the cell region is inclined; *Non-ground:* the cell has an obstacle or overhang or is lower in height (drop-off) than nearby ground regions; *Unknown:* there is insufficient or no information about the region.

This safety map is then used by the Local Metrical level as its LPM, by having *Level* and *Inclined* cells in the safety map correspond to free space in the LPM and *Non-ground* cells correspond to obstacles. For a 3D grid 14x14x3 meters in size, with 10 cm resolution, the current implementation can update an LPM at $\sim$4 Hz.

Assumptions

We use the horizontal lidar on the wheelchair robot to keep the robot localized with respect to a lidar-based LPM. This is done using a 3-DOF SLAM algorithm. Visual SLAM techniques are currently too computationally intensive to run concurrently with the 3D modeling and traversability abstraction. Therefore, for the experiments reported in this paper our robot is restricted to traveling only on near-level surfaces.

In the future we intend to replace 3-DOF method with a camera-based 6-DOF SLAM algorithm [3]. The 3D modeling algorithm is general and applicable without modification to the case when the robot knows its 6-DOF pose in the local 3D model.

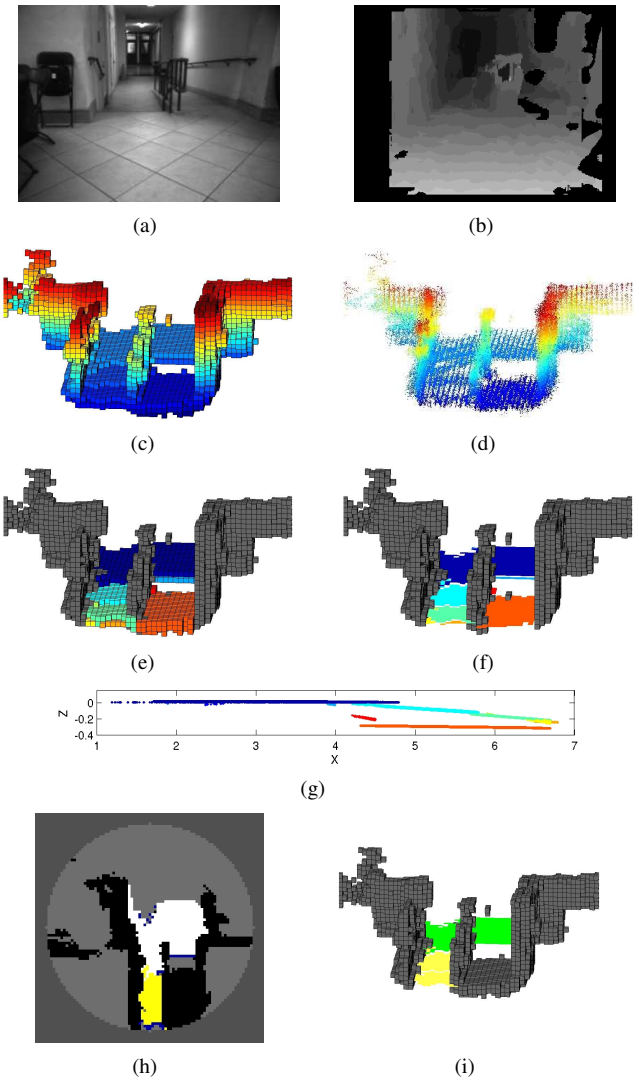


Fig. 4. (Best viewed in color) (a) Left image from the stereo camera showing a typical non-planar scene, with a drop-off to the left, and a ramp to the right, of the railing. (b) Disparity map computed for the stereo pair (brighter regions closer). (c) 3D occupancy grid of the above environment (voxels colored according to height) constructed from a collection of stereo images. (For clarity, the viewpoint of this figure is different from the viewpoint of the left image above.) (d) 3D point cloud that built the occupancy grid. (e) Potentially traversable ground regions found by segmenting the 3D grid. (Obstacles are gray.) (f) The planes obtained for each of the potential ground segments. (g) A cross-sectional view of the planes. (h) The final safety map obtained after analyzing the segments and fitted planes for safety: black for *Non-ground* regions; white for *Level* regions; yellow for *Inclined* regions; light gray for *Unknown* regions; dark gray for unexamined regions; and blue for denoting *Potential Drop-off Edges* that might be present in the *Level* and *Inclined* regions. (i) A hybrid 3D model can also be constructed at this point. The planes are used to represent traversable ground regions (green for *Level* and yellow for *Inclined* regions). The grid is used to represent *Non-ground regions* that are not modeled easily using planes.

IV. Comfortable Motion for a Wheelchair

A robot transporting a human passenger not only needs to plan obstacle-free paths, but it also needs to compute how to move on the path such that the motion is comfortable. That is, it needs to find a *trajectory*—a time parameterized function of robot pose. Below we give an overview of our formulation of trajectory planning as a variational minimization problem (described and quantitatively evaluated in previous work [8]). We then discuss particular issues in integrating this work with the existing HSSH path planner.

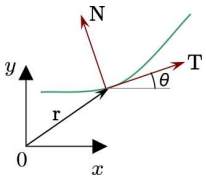


Fig. 5. Tangent and Normal to a curve.

Given boundary conditions on pose, velocity, and acceleration at both end-points, our objective is to find a trajectory that satisfies the boundary conditions and minimizes the discomfort. The discomfort is modeled by a cost functional J, which is a function of the total travel time and motion as parameterized by time.

For a robot moving on a planar curve, $\mathbf{r}(t) = (x(t), y(t))$ denotes the position vector at time t. The unit tangent and normal vectors to the curve are given by $\mathbf{T}$ and $\mathbf{N}$ respectively. The angle θ that the tangent makes with the x axis is given by: $\theta = \text{atan2}(\dot{y}, \dot{x})$. The robot is modeled as a rigid body moving in a plane subject to the nonholonomic constraint $\dot{x}\sin\theta - \dot{y}\sin\theta = 0$. To ensure that this constraint is satisfied, we assume that the x axis of the body-centered coordinate frame is always tangent to the curve $\mathbf{r}(t)$.

The discomfort measure is the following cost functional:

$$J = \tau + w_T \int_0^\tau (\dddot{\mathbf{r}} \cdot \mathbf{T})^2 \, dt + w_N \int_0^\tau (\dddot{\mathbf{r}} \cdot \mathbf{N})^2 \, dt + w_{\dot{\theta}} \int_0^\tau \dot{\theta}^2 \, dt + w_{\ddot{\theta}} \int_0^\tau \ddot{\theta}^2 \, dt$$

τ represents the total travel time, and $\dddot{\mathbf{r}}$ represents the jerk. $\dddot{\mathbf{r}} \cdot \mathbf{T}$ and $\dddot{\mathbf{r}} \cdot \mathbf{N}$ are the tangential and normal components of jerk respectively. $\dot{\theta}$ is the angular velocity, and $\ddot{\theta}$ is the angular acceleration. We assume that $\mathbf{r}(t)$ is smooth enough for the cost functional to be well-defined. This means that the acceleration vector is continuous and normal and tangential components of jerk are square integrable.

The term τ is necessary. If it is not included in the functional, the optimal solution is to reach the destination at $\tau = \infty$ traveling at essentially zero speed in the limit (except perhaps at the end-points where the speed is already specified). Thus, minimizing just the integral terms will not lead to a good solution.

The weights (w_T, w_N, $w_{\dot{\theta}}$, $w_{\ddot{\theta}}$) are non-negative, real numbers. The weights serve two purposes. First, they act as scaling factors for dimensionally different terms. Second, they determine the relative importance of the terms. The weights provide the ability to adjust the performance according to user preferences. For example, on a wheelchair, some users may not tolerate high jerks and prefer traveling slowly while others could tolerate higher jerks if they reach their destination quickly. The weights are determined via dimensional analysis of the cost functional so that discomfort is independent of boundary conditions. For this work, we utilize the "characteristic weights", which were previously determined [8].

The optimization problem is to find a function $\mathbf{r}$ and a scalar τ that minimize J given the boundary conditions:

$$\begin{aligned} \mathbf{r}(0) &= \mathbf{r}_0, & \mathbf{r}(\tau) &= \mathbf{r}_\tau, \\ \theta(0) &= \theta_0, & \theta(\tau) &= \theta_\tau, \\ \dot{\mathbf{r}}(0) &= v_0 \mathbf{q}_0, & \dot{\mathbf{r}}(\tau) &= v_\tau \mathbf{q}_\tau, \\ \ddot{\mathbf{r}}(0) \cdot \mathbf{T}(0) &= a_{T_0}, & \ddot{\mathbf{r}}(\tau) \cdot \mathbf{T}(\tau) &= a_{T_\tau}. \end{aligned} \quad (1)$$

Here $\mathbf{q}_0 = (\cos\theta_0, \sin\theta_0)$ and $\mathbf{q}_\tau = (\cos\theta_\tau, \sin\theta_\tau)$, v is the speed and a_T is the tangential acceleration. In the following discussion, the subscripts T and N stand for the tangential and normal components of a quantity respectively.

The variational optimization problem of Equation 1 is posed in an infinite dimensional space of vector-valued functions $\mathbf{r}(t)$. We minimize J in a finite-dimensional subspace by discretizing $x(t)$ and $y(t)$.

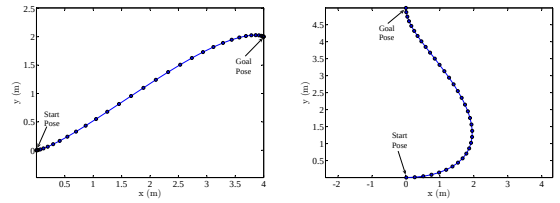
For J to be well-defined in this subspace, θ and its first and second derivatives need to be well-defined. θ is not an independent variable but is determined by $\theta = \text{atan2}(\dot{y}, \dot{x})$ when the tangential speed is non-zero. Different expressions for θ have to be derived when the tangential speed is zero. For the robot to move in the "forward" direction, the speeds v_0 and v_τ should be non-negative. Since the optimal trajectory tries to keep the travel time small, it is clear that for the optimal trajectory the tangential speed will never be zero. Thus, θ will always be well-defined in the interior $(0, \tau)$. The only trouble can arise at the two end-points, where the specified tangential speed may be zero. Previous analysis [8] shows that there are two types of boundary conditions where speed is zero: (i) $v = 0, a_T \neq 0$, (ii) $v = 0, a_T = 0$. For the second type of boundary condition, the expression for θ can be specified in terms of the third derivatives of $x(t)$ and $y(t)$. Thus, for θ to be well-defined, the discretization of $x(t)$ and $y(t)$ should be such that their third-derivatives exist.

Thus, in order to completely define the problem we need to specify 4 boundary conditions per end-point per space dimension—position and three derivatives. Hence, we choose heptic interpolating splines as the basis functions. Heptic splines are degree seven piecewise polynomials with continuous derivatives up to order six. As a function, each spline $x(t)$ and $y(t)$ ($M+1$ polynomial pieces) can be uniquely determined from 8 boundary conditions and its value on M interior nodes. In addition to the travel time τ, these nodal function values $\{x_i, y_i\}_{i=1}^{M}$ are the parameters that are found by optimization. In the input specification of Equation 1, only derivatives of up to second order (position, velocity and acceleration) are given. The values of normal acceleration a_N, tangential jerk j_T, and normal jerk j_N are left as unknown parameters for the optimization problem. These are determined along with the optimal trajectory.

Figures 6(a) & 6(b) illustrate the paths corresponding to the optimal trajectory for two cases with different boundary conditions.

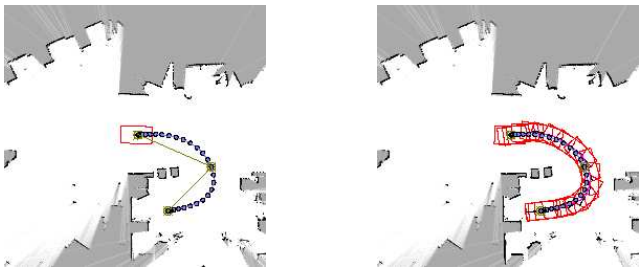
Avoiding Obstacles

Above, we discussed an algorithm for generating trajectories between an initial and a final pose, given the velocity and acceleration at both end-points, such that the resulting motion is comfortable for a human passenger. Noticeably lacking is any notion of safety. As part of the integration task, we combine the existing HSSH Local Metrical planner together with the above algorithm to compute safe trajectories. The result is a geometric path that (in practice) does not intersect obstacles, while the motion on the path is comfortable.



(a) (b)

Fig. 6. Optimal paths for two examples. The circles are drawn at equal intervals of time; thus, lesser spacing between circles implies higher speed. (a) Start pose $(x, y, \theta)_0 = (0, 0, 0)$, End pose $(x, y, \theta)_\tau = (0, 0, -\pi/4)$. Velocity and acceleration at both ends are zero. The boundary conditions on orientation can be imposed at end-points even when speed and acceleration are both zero. As expected, the path is almost a straight line. The robot starts moving slowly, accelerates to maximum velocity, and then slowly comes to a stop. (b) Start pose $(x, y, \theta)_0 = (0, 0, 0)$, End pose $(x, y, \theta)_\tau = (0, 5, \pi/2)$. The initially velocity is 1 m/s to the right. The normal and angular jerk terms in J ensure that the robot does not turn too fast resulting in a gently curved path.



(a) (b)

Fig. 7. (a) A real world example of a comfortable trajectory. This trajectory is composed of several sub-goals, given by a trivial RRT planner. (b) Actual path of the robot. A static feedback linearization controller [10] is used to compute the control commands necessary to follow the trajectory.

At the Local Topology level, the robot uses the forward-facing gateway (and the underlying *Voronoi skeleton* used to find gateways [1]) to continually chose a new goal point at the edge of the LPM. This facilitates navigation down hallways. At places, the gateways themselves are used as goals to facilitate large-scale turn actions. At the Local Metrical level, the driver may click a position to travel to in the LPM. The integration task here is to transform the continually computed goal locations into safe and comfortable trajectories from the robot's current location.

The HSSH utilizes an efficient Rapidly-expanding Random Tree (RRT) [9] planner (see Figure 7(a)) to compute piecewise linear plans from goal points. Given a plan of safe waypoints, a trajectory must be computed. The boundary conditions are: zero velocity and acceleration at the goal point, the robot's current velocity and acceleration at the start point, and a specified velocity at all intermediate points. In the current implementation, the magnitude of this velocity is specified as the desired average speed of the wheelchair; however, in future, the boundary conditions at the intermediate points will be found by optimization. Figure 7 shows a path corresponding to an optimal comfortable trajectory. The piecewise linear path produced by the RRT planner is also shown. The RRT planner is capable of running very fast, but is only rerun as the LPM is updated (often 10 Hz with a lidar-based LPM). A trajectory can be computed from a plan at $\sim$5 Hz.

In theory, this method does not ensure trajectories that

completely avoid obstacles. However, in practice, we rarely see the optimal trajectory come too close to an obstacle. When it does, the robot's control avoids collisions, and a new plan ultimately moves the robot away from the obstacle.

V. Integration Progress and Results

In this section we show that the 3D depth information from a stereo camera can reliably disambiguate between drivable surfaces and non-traversable stairs or curbs in indoor and certain outdoor environments. We illustrate situations where the vision-based Local Perceptual Map (LPM) is safer than lidar-based models, though occasionally stereo vision fails to detect textureless surfaces. We also demonstrate the local topology and trajectory generation algorithms working successfully with the vision-based LPM.

The vision-based LPM we use in our system is 14 meters wide with a cell resolution of 10 cm, resulting in a 140x140 cell grid. In order for the full system to run smoothly and reliably, components cannot run at full speed, even on modern multiprocessor machines. As such, we throttle the system components: the vision-based LPM is updated at $\sim$2 Hz (the lidar-based LPM that is currently used for localization is run synchronously); the gateways, local topology, and travel goal points are updated at $\sim$1 Hz; thus, new paths and trajectories to LPM goal points are generated at $\sim$1 Hz.

A. Integration Successes

The new HSSH implementation that integrates stereo vision LPMs and comfortable trajectories has shown promising results in various situations that were not well handled in previous HSSH implementations.

Figure 8 illustrates how the vision-based LPM finds different places than a lidar-based LPM. The wheelchair is in a large region that is basically a large + shaped intersection with curved walls and a circular railing in the middle with stairs. When using a lidar-based LPM, the robot will hypothesize a single place with gateways at the edges of the actual hallways, a (potential) + intersection. This is because the gateway algorithm removes "island" obstacles (in this case the thin railings) from the LPM prior to its analysis of the local structure. The vision-based LPM clearly detects the thin metal rails as belonging to single a continuous obstacle, and instead parses the large region into a set of smaller places.

At first, the HSSH local topology algorithm hypothesizes a potential place with four ways out using the vision-based LPM (see Figure 8(a)). Before deciding that it really is at a place, the robot moves to a point near the center of the place neighborhood and spins around to get more information.[1]

Upon moving closer to the stairwell, the robot detects the drop-off (see Figure 8(b)); thus, an obstacle is created at this location in the 2D LPM representation of the local region. Consequently, the gateway algorithm finds only three gateways (see Figure 8(c)), which align to form a circular ordering corresponding to a Y intersection. The wheelchair decides that this is definitely a place and waits for the user to select the gateway to leave through.

[1]The idea of rotating in place as exploration of a potential place is historical. It works well with a small, circular robot but is not ideal for a robot with a human passenger. This will be addressed in future work.

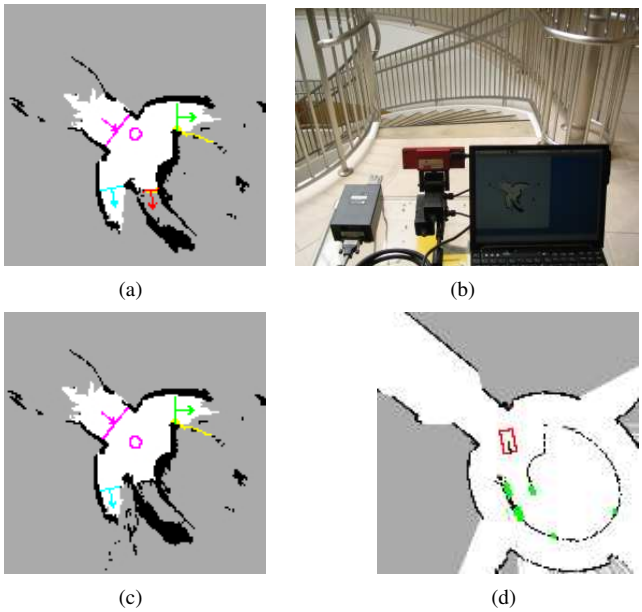


Fig. 8. (a) The robot begins mapping a large open intersection. Using the vision-based LPM, it hypothesizes four ways out of the current region. (b) Upon further examination, a drop-off due to a downward stairwell is detected. (c) The robot verifies that it is indeed at a topological place; however, the final symbolic local topology describes a simpler Y intersection, with only three (safe) ways out—the red arrow in image (a) corresponded to the drop-off. (d) The lidar-based LPM does not see the drop-off, which could be catastrophic. (The green blobs represent dynamic obstacles, which occur due to the lasers not seeing the poles consistently.)

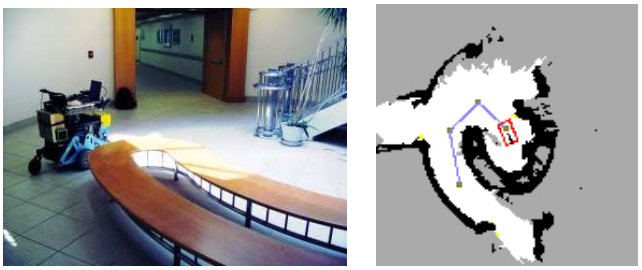
Figure 9 illustrates the successful integration between the trajectory planner and the vision-based LPM—the trajectories are similar to those obtained with the lidar-based map in Figure 7. The vision-based LPM allows the robot to avoid the bench (shown in Figures 9(a) & 9(d)) that has an overhang that is too high to be seen by the horizontal lidar.

Figure 10(a) shows an outdoor area that the wheelchair can only navigate using the vision-based LPM. The 3D hybrid model detects a difference between the sidewalk height and height of the road to the left. The retaining wall on the right is also easily detected. These height differences appear as obstacles in the LPM, and the Local Topology level of the HSSH detects this as a single path. The robot chooses a goal point ahead on the path (green dot on the right of Figure 10(b)), and plans a trajectory. Figure 10(d) shows that the cars to the left create obstacles even in the lidar-based LPM. However, at empty parking spots, the lidar-based LPM creates large regions of free space that lead to false positive detections of L intersections, and (as in Figure 8(d)) provide the robot with an unsafe model of the local surround.

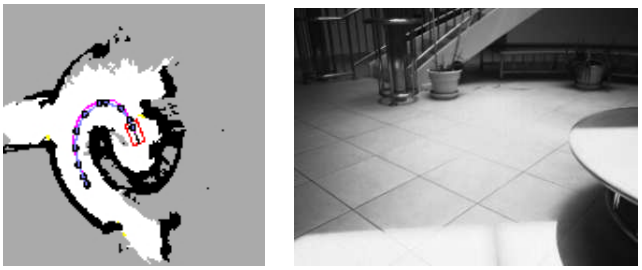
B. Integration Drawbacks/Failures

Despite the successes discussed above, there are certain limitations of each component discussed in this paper. Some of these only become obvious upon integration into a larger system and lead to novel problems to tackle in future work.

One issue is the amount of stereo vision data needed to build the hybrid 3D model (due to the noisy nature of



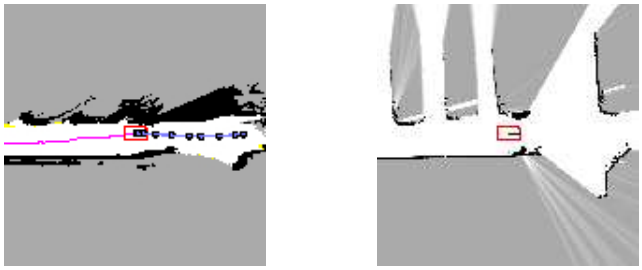
(a) (b)



(c) (d)

Fig. 9. (a,b) The stereo camera is able to detect an overhanging bench top that cannot be seen by the wheelchair's horizontal lidar. Thus, in this scenario the vision-based LPM provides a useful model for safe planning. (c) The wheelchair computes a trajectory that results in a smooth path and comfortable motion by using the vision-based LPM. (d) A snapshot from the robot's camera as it navigates around the bench.

(a) (b)



(c) (d)

Fig. 10. (a) The wheelchair navigates down a cluttered sidewalk. It senses a drop-off (and cars) on the left and a small wall on the right. (b) These changes in height create boundaries in the LPM that allow the Local Topology level to easily determine gateways that define a path and an aim point ahead along the path. (c) The trajectory is such that the wheelchair comes to rest at the goal. (d) The lidar-based LPM does not see the drop-off, and cannot be used in these situations.

stereo data). This can be seen by comparing the LPMs in Figures 8(c) and 8(d), generated from the exact same motion of the robot. The vision-based LPM cannot adequately model the environment beyond about 4 meters whereas the lidar sensor can detect obstacles up to 80 meters. This affects the speed at which the wheelchair can drive, as it needs to move slow enough to reliably detect the ground, obstacles, and, drop-offs, etc. It also means dynamic obstacles are generally undetected, which is why slow re-planning (at 1 second intervals) is currently acceptable.

Vision also requires good lighting to work properly. In poor lighting, surfaces lose texture and the stereo camera has difficulty computing disparity information. Figure 11 shows a situation where the robot is navigating a hallway and turns into a hall with low lighting. As it approaches unknown (gray) space in the LPM, the lack of depth information about the floor means that the safety properties of this region remain unknown. The detected local topology represents a dead end (Figure 11(b)). Because of this, the robot does not attempt to drive over unknown terrain (an invisible floor appears the same as a bottomless pit in the vision-based LPM). This is a useful feature of the integrated system.

Low textured environments are also problematic for stereo vision due to the lack of salient features. Figures 12(a) & 12(b) show a common situation where a featureless wall leads to a (false negative) region of no obstacles in the LPM. Free space (corresponding to the ground) is next to unknown space in the LPM. The gateway algorithm sees this as an opening to be explored, and a false positive place is generated with a T local topology structure. One possible method to handle this is to put *virtual* obstacles at unknown cells in the LPM that border free cells. However, this creates

(a) (b)

Fig. 11. Low light causes floors to appear texture-less leading to poor stereo distance information. (a) The robot arrives at the T intersection in Figure 12(a). The right (downwards facing) hallway is poorly lit. (b) As the wheelchair travels down this hallway it quickly arrives at the "frontier" of free space since the dark floor remains unseen (and incorrect distance information leads to phantom obstacles). This results in no gateways facing down the hall causing the wheelchair to believe it is at a dead end.

obstacles at the true frontiers of experience and at real-world occlusions, inhibiting the gateway algorithm from working at all. Figures 12(c) & 12(d) show an extreme example of a textureless wall immediately outside our robot lab.

In addition to the perceptual issues above, there are several planning and control issues. The vision-based LPMs are noisier than lidar-based LPMs and as a result, narrow hallways and paths that the wheelchair could navigate when using a lidar map, do not yield safe paths in the fuzzier vision-based LPMs.

In traveling down hallways, the robot uses the forward-facing gateway to continually chose a new goal point at the edge of the LPM. In curved hallways, the RRT plan can be quite different for each new goal point. Since the trajectories are dependent on the nodes of the RRT plan, this can lead to large changes in the robot's heading at the start of a new

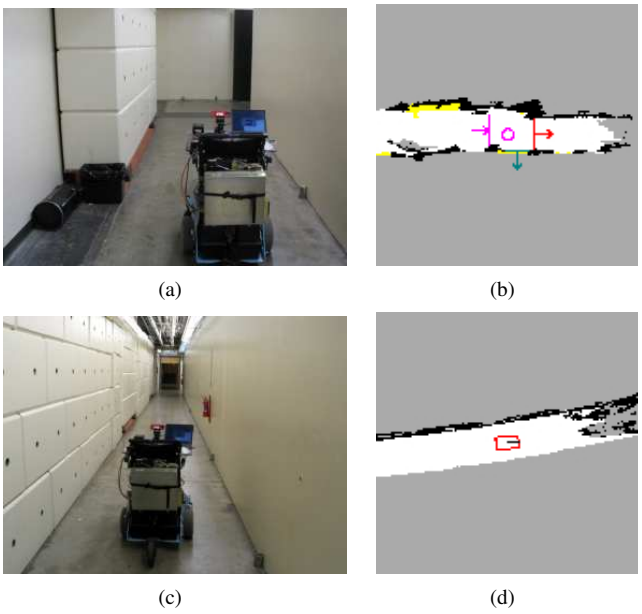


Fig. 12. (a) Texture-less walls (e.g., the solid wall at the right of this image) often lead to poor distance information from the stereo camera. (b) This causes gaps in the wall when it is modeled in the vision-based LPM. This leads to incorrect gateways and to false positive place detections in the environment. (c) A particular wall that is often completely invisible to stereo. (d) This leads to an LPM with a large boundary between free and unknown regions, which results in no gateways being found. Thus, the wheelchair has no way to autonomously travel down this hallway in the current implementation.

trajectory. The result is that the robot's heading noticeably oscillates as it moves down the hallway. This can be fixed by using a slower, but more stable planner in the future.

VI. Conclusion

This paper demonstrated successful integration of the three technologies needed for an inexpensive, usable robotic wheelchair: comfortable motion generation, safe models of common non-planar situations from vision sensing, and natural, infrequent navigation commands. The 3D hybrid model created purely from stereo vision (assuming accurate localization) is sufficient for safe planning in environments with potentially dangerous drop-offs, overhanging obstacles, or ramps. Trajectories can be computed on top of safe plans, which result in motion that minimizes the discomfort of the human passenger. On top of this, the HSSH framework provides place detection, qualitative descriptions of the place structure, and a causal interface for large-scale commands. Although we have discussed some integration problems (mainly due to perception issues), we feel these are solvable in the near future.

The integration demonstrated here is only an initial stage towards a complete implementation of the Intelligent Wheelchair—an intelligent mobility aid for people with mobility, perception, communication, and cognitive disabilities. Though the wheelchair is specifically aimed towards disabled users, we envision this technology generalizing to personal transports of various sizes and domains, used by large portions of future populations.

Future Work

The integration process and our results show several directions for further work. The most obvious direction is the need to improve the computational efficiency of the vision-based LPM. Another important problem is that of low texture. We want to develop an algorithm that distinguishes between (and annotates) true unknown space in the visual LPM and unknown space arising due to low texture. This will allow the local topology algorithm to treat unknown cells arising due to low texture as virtual obstacles when finding gateways. A longer term solution is to use color models and/or other image features to hypothesize disparities in low texture regions.

A problem of more immediate importance is accounting for obstacles when generating trajectories. It might be possible to include obstacles as constraints in the optimization formulation for trajectory generation allowing for seamless integration with the current system. Other pieces of future work include: detecting and describing outdoor places not defined by path boundaries, using color and texture in addition to geometry to determine traversability, designing an intuitive user interface for tuning comfortable motion parameters, and full integration with the HSSH global topological and metrical levels.

ACKNOWLEDGMENTS

The authors wish to thank Joseph Modayil, who contributed significantly to the HSSH control infrastructure, and Chetan Jhurani, who contributed significantly to the trajectory planning framework.

References

[1] P. Beeson, "Creating and utilizing symbolic representations of spatial knowledge using mobile robots," Ph.D. dissertation, The University of Texas at Austin, 2008.

[2] P. Beeson, J. Modayil, and B. Kuipers, "Factoring the mapping problem: Mobile robot map-building in the Hybrid Spatial Semantic Hierarchy," *Int. Journal of Robotics Research*, in press. [Online]. Available: http://ijr.sagepub.com/cgi/content/abstract/0278364909100586v1

[3] A. I. Comport, E. Malis, and P. Rives, "Accurate quadrifocal tracking for robust 3D visual odometry," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 40–45.

[4] B. Kuipers, "The Spatial Semantic Hierarchy," *Artificial Intelligence*, vol. 119, pp. 191–233, 2000.

[5] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 321–328.

[6] A. Murarka and B. Kuipers, "A stereo vision based mapping algorithm for detecting inclines, drop-offs, and obstacles for safe local navigation," in *Proc. of the IEEE/RSJ Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[7] K. Konolige and D. Beymer, *Small Vision System v3.2*, Videre Design LLC, November 2004.

[8] S. Gulati, C. Jhurani, B. Kuipers, and R. Longoria, "A framework for planning comfortable and customizable motion of an assistive mobile robot," in *Proc. of the IEEE/RSJ Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[9] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An efficient approach to single-query path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San Francisco, California, April 2000, pp. 995–1001.

[10] A. D. Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control*, ser. Lecture Notes in Control and Information Sciences, J. P. Laumond, Ed. New York: Springer-Verlag, 1998, pp. 171–264.