Abstract—This paper presents a navigation method for a mobile robot in urban environments. The navigation method employs a so-called Polar Traversability Index (PTI) to evaluate terrain traversal property along a specific direction and guide the robot’s motion. In the navigation system, a 2-D Laser Range finder (LRF) is used to produce terrain maps in various formats, such as 3-D point-cloud map, 2.5 D elevation map and terrain traversability map. In order to generate a reliable elevation map for navigation, a filtering method based on the robot motion constraint and the LRF’s characteristics is used to filter out erroneous range data. The elevation map is converted into a one-dimensional PTI histogram where each PTI dictates the level of difficulty for the robot to traverse the terrain in the corresponding direction. The PTI histogram is then used to determine the robot’s motion commands—speed and steering rate. The PTI histogram enhances the robot’s navigational capability in urban environments. For instance, it enables the robot to traverse wheelchair ramps and avoid curbs when negotiating sidewalks and it allows the robot to sink through a ramp which may be too steep to travel for a robot using existing traversability measures. The efficacy of PTI has been verified by simulation and experiments in a complete navigation system.

Index Terms—obstacle negotiation, obstacle avoidance, terrain traversability analysis, traversability index, traversability map, polar traversability index, polar traversability histogram.

1. INTRODUCTION

Most of the existing research on autonomous navigation of mobile robot in urban environments has been mainly focused on lane/curb detection and tracking [1, 2, 3]. Many of these systems [1, 2] use a video camera and employ image processing techniques to extract lane/curb features. For an instance, Lai and Yung [1] proposed a multiple lanes/curbs detection method based on orientation and length features of lane marking and curb structures. Curb detection based on stereovision [4] is advantageous over the above methods as it provides 3-D information of the curb. However, systems using video camera(s) are sensitive to illumination condition. To overcome this disadvantage, Wijesoma et al. [3] used a 2-dimensional (2-D) Laser Rangefinder (LRF) for curb detection and tracking. Their method first segments the laser range data points using an Extended Kalman Filter and extracts curb segments based on curb features known ahead of time. The system then tracks the curb and allows the robot to follow the curb.

This paper targets at a different way of guiding a mobile robot in an urban environment which is considered as one with hard surfaces and comprising curbs, ramps and obstacles. An indoor environment with flat ground, steps and wheelchair entries is also considered as urban terrain in this paper. The task of navigating a mobile robot in an urban environment is to steer the robot in such a way that it avoids a curb that blocks its way but traverses low-profiled obstacle(s) and ramp(s) as necessary (e.g., moves up to a sidewalk via a wheelchair ramp). This ability is called Obstacle Negotiation (ON) in the literature. An essential component in an ON system is terrain mapping [5]. Stereo vision [5, 6, 7] and 3-D LRFs [8, 9] have been widely used for this purpose. Stereo vision is sensitive to illumination condition and has low range resolution and accuracy; while 3-D LRFs are costly and have slow frame rates that may affect the robot’s mobility. The author of this paper developed a cost-effective terrain mapping method [10] using a 2-D LRF which had been tested in a rotary table in an offline manner. In this paper, the mapping method is extended to its online version and tested with a robot with 6 degree-of-freedom motion.

Another important module in an ON system is Terrain Traversability Analysis (TTA) which analyzes the traversal property of a terrain map. The TTA module usually transforms an elevation map into a traversability map which is then used to plan the robot motion. A representative TTA method is proposed by Gennery [11] which fits a least-square plane to a small terrain patch and uses the roll and pitch of each fitted plane, along with the residual of the fit to estimate the slope and roughness of the terrain patch. To cope with the inaccuracy of the stereovision data, the algorithm computes the covariance matrix of each data point and estimates the slope iteratively. It is therefore computationally complex. An alternative TTA method is to use the Traversability Index (TI) [6, 16, 17, 18] to describe how easily the robot traverses a terrain cell. The TIs in [6] are represented by a number of fuzzy sets. In this paper, a new measure—Polar Traversability Index (PTI)—is proposed to evaluate terrain traversability along a specific direction. A PTI represents the level of difficulty to traverse along the corresponding direction and it allows a robot to snake up/down a steep ramp.

This paper is organized as follows: In Section II the robot is introduced for urban navigation. In Section III a brief overview of the ON system is provided. Then, in Sections IV and V the terrain mapping and traversability analysis algorithms are detailed. In Section VI the PTI concept is
presented. The simulation and experimental results are given in Section VII and the paper is concluded in Section VIII.

2. THE ROBOT PLATFORM FOR NAVIGATION

A pioneer robot P3-DX [12] as depicted in Fig. 1 is used in this work. The robot has a differential drive mechanism and can turn with a zero turning radius. Since the robot moves on uneven terrain, its roll, pitch and yaw angles (Euler angles) must be sensed for position estimation and map-building. This is achieved by a 3-axis fiber optic gyro. The robot’s roll, pitch and yaw angles, and its X, Y, and Z coordinates (called robot pose collectively) are then used to register the laser range data into a terrain map.

The on-board computer uses a 2.0 GHz Pentium 4 processor. It acquires the laser range data from the LRF (Sick LMS 200) at 500k baud and the robot Euler rate from the gyro at 38400 baud through a RS-422 card. It obtains the robot’s wheel encoder data at 115200 baud through a regular RS-232 port. These data are then used for terrain mapping and navigation. The robot motion commands are determined by the ON software based on the terrain map and used to control the robot. The onboard computer also transmits the robot pose and laser range data to an off-board laptop computer via the wireless router. The process of terrain mapping and navigation is thus visualized on the laptop computer in real-time.

3. OVERVIEW OF THE NAVIGATION SYSTEM

As depicted in Fig. 2, the navigation system consists of three main modules: Terrain Mapping, Terrain Traversability Analysis (TTA) and Path Planning. The vision sensor for terrain mapping is the LRF (SICK LMS 200) which is mounted on the front-end of the robot. The LRF looks forward and downward at the terrain with an angle of -10° from the robot base plate. While the robot is in motion, the fanning laser beams profile the terrain ahead of the robot and produce a terrain map. In our system, a 2-D grid-type terrain map is used for ON.

The TTA module transforms the terrain map into a traversability map where each cell holds a value representing the degree of difficulty for the robot to move across that cell. This value is called Traversability Index (TI). The Path Planning module converts the traversability map into a number of PTIs each of which represents the level of difficulty for the robot to move along the corresponding direction. Based on the PTIs, the Path Planning module determines the steering and velocity commands of the robot and sends these motion commands to control the robot.

4. TERRAIN MAPPING AND TERRAIN MAP

Terrain mapping is a process of transforming laser range measurements to 3-D points in the world coordinate (also called navigational frame) and registering these data points into a map. In this work, an Extended Terrain Map (ETM) is first built. Each ETM consists of an elevation map (also called terrain map) and a so-called “certainty map.” Both are 2-D grid-type maps. Each cell in the elevation map holds a value representing the height of the object at that cell; while each cell in the certainty map holds a value representing the certainty that the corresponding cell in the elevation map is occupied by the object.

Homogeneous coordinate transformation is used to convert a range measurement into a point in the navigational frame. Figure 3 depicts the coordinate systems used for terrain mapping. The left figure depicts the diagram of the robot and the right figure details the coordinate systems fixed on the robot body. In Fig. 3, x₀ y₀ z₀ is the robot body frame fixed to the midpoint of the wheel axel. Frames x₁ y₁ z₁ and x₂ y₂ z₂ are the coordinate systems fixed to the LRF’s mounting bracket and the LRF, respectively. The origin of frame x₁ y₁ z₁ is at (0, m, g+h) in frame x₀ y₀ z₀. The angle between z₁ and z₀ is α which is the LRF’s initial tilt-down angle, i.e., -10°. Frame x₂ y₂ z₂ is the coordinate system attached to the laser receiver and its origin locates at (0, b, p) in frame x₁ y₁ z₁. The navigational frame x₃ y₃ z₃ is aligned with the robot body frame when a navigation task is specified and it remains constant throughout the entire navigation task. The origin of the robot body frame is at (u, v, w) in the navigational frame. The three Euler angles are defined as follows: roll (φ), pitch (θ) and yaw (ψ) which are the rotation angles around y₁, x₀, and z₀ axes, respectively.
The homogeneous transformation matrix $T^n_b$ can be derived through successive rotations and translations from frame $x_b,y_b,z_b$ to frame $x_n,y_n,z_n$. The coordinates of a point in frame $x_b,y_b,z_b$ can be converted into the coordinates of a point in frame $x_n,y_n,z_n$ by applying a roll rotation $Rot_x(\theta)$, a yaw rotation $Rot_y(\psi)$ and a translation $Trans(u,v,w)$. The corresponding transformation matrix $T^n_b$ is given by

$$T^n_b = Trans(u,v,w)Rot_y(\psi)Rot_x(\theta)Rot_z(\phi) = \begin{bmatrix} (c \psi c \phi - s \psi s \phi) & -s \psi c \phi + c \psi s \phi & u \\ (s \psi c \phi + c \psi s \phi) & c \psi c \phi - s \psi s \phi & v \\ -c \psi s \phi & s \phi & c + w \end{bmatrix}.$$  (1)

where $c\phi$ and $s\phi$ stands for $\cos(\phi)$ and $\sin(\phi)$, respectively. Similarly, the transformation matrix that converts the coordinates of a point in frame $x_b,y_b,z_b$, into the coordinates of a point in frame $x_b,y_b,z_b$ is

$$T^b_r = Trans(0,m,g + h)Rot_x(\alpha)Trans(0,b,p) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & -s\alpha & hbc\alpha - psa + m \\ 0 & s\alpha & c\alpha & bsa + pca + g + h \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$  (2)

Therefore, $T^n_r$ is computed by

$$T^n_r = T^n_bT^b_r = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$  (3)

A range measurement $l$ is converted into a 3-D point $\vec{d}^r = (l\cos\beta,l\sin\beta,0)^T$ in frame $x,t,y,z$, where $\beta = n\pi/180$ is the scanning angle of the $n^{th}$ laser measurement. Transforming $\vec{d}^r$ into a point $(x_n, y_n, z_n)^T$ in the navigational frame produces

$$\vec{d}^n = T^n_r\vec{d}^r = \begin{bmatrix} q_{11}l\cos\beta + q_{12}l\sin\beta + q_{14} \\ q_{21}l\cos\beta + q_{22}l\sin\beta + q_{24} \\ q_{31}l\cos\beta + q_{32}l\sin\beta + q_{34} \\ 1 \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}.$$  (4)

For every two consecutive measurements (at time steps $t$ and $t+1$) on the object at the same location $(x_n, y_n)$, the expected change in height value $\Delta z$ should not be bigger than

$$\Delta z_{\text{max}} = \Delta u \tan \phi - \Delta v \tan(\theta + \alpha) + \Delta w + \frac{\partial z_n}{\partial \psi} \Delta \psi,$$

$$+ \frac{\partial z_n}{\partial \theta} \Delta \theta + \frac{\partial z_n}{\partial \phi} \Delta \phi + \frac{\partial z_n}{\partial \psi} \Delta \psi$$  (5)

where $\Delta$ is the maximum measurement error of the LRF (32 mm according to [10]); $\Delta \theta$, $\Delta \phi$, and $\Delta \phi$ are the changes in the robot’s pose from $t$ to $t+1$. $\partial z_n/\partial \psi$, $\partial z_n/\partial \theta$, $\partial z_n/\partial \phi$ and $\partial z_n/\partial \psi$ are the partial derivatives and they can be derived from (4).

In this paper, a grid cell size of $80 \times 80$ mm is used. The height value of cell $(i,j)$ in the elevation map is denoted by $h_{i,j}$, while the certainty value of this cell in the certainty map is denoted by $c_{i,j}$. Using the above transformation, a range data is translated into a point $(x_n, y_n, z_n)$. In order to register this point into the map at the associated location, the coordinate values $x_n$ and $y_n$ are converted to the grid indices $i$ and $j$, respectively. Then the certainty value $c_{i,j}$ and height value $h_{i,j}$ of the cell in the terrain map are updated. Specifically, $c_{i,j}$ is updated by

$$c_{i,j}^{t+1} = \begin{cases} c_{i,j} + a & \text{if } |z_n^{t+1} - h_{i,j}^{t+1}| \leq \Delta z_{\text{max}} \text{ or } c_{i,j}^{t+1} = 0 \\ c_{i,j}^{t+1} & \text{otherwise} \end{cases}.$$  (6)

where $a \geq 1$ is the increment of the certainty value and it can be any positive integer. In this study the certainty value is represented by one byte and $a=3$ is used. As a result, all cells illuminated more than 84 times have the same certainty value. The height value $h_{i,j}$ of the cell in the elevation map is updated by

$$h_{i,j}^{t+1} = \begin{cases} z_n^{t+1} & \text{if } z_n^{t+1} > h_{i,j}^{t} \text{ or } c_{i,j}^{t+1} = 0 \\ h_{i,j}^{t} & \text{otherwise} \end{cases}.$$  (7)

$c_{i,j}$ and $h_{i,j}$ are initially zero and updated by (6) and (7), respectively. In every 13.3 milliseconds the 181 range measurements of the LRF are mapped into both maps. In (6), the condition $|z_n^{t+1} - h_{i,j}^{t}| \leq \Delta z_{\text{max}}$ is called motion continuity constraint [10]. Every two consecutive measurements on a real object at the same location should
satisfy this constraint. According to (6), cells occupied by real object are assigned continuous certainty increments and result in large certainty values. Therefore, they can be identified easily. Mixed pixels and random noise, on the other hand, don’t satisfy the motion continuity constraint and thus result in small certainty values. In addition, they are spatially isolated in the elevation map. These characteristics can be used to filter out mixed pixels and random noises. Details of the filtering method are referred to [10]. In [10], the filtering method is used in an off-line manner, i.e., it was applied after the elevation maps were built. In this work, the filtering method is applied to filter out the mixed pixels and random noise in the laser data on-the-fly. (Missing data is not likely encountered in this outdoor scenario as the roughness of the terrain surfaces is big.)

5. Terrain Traversability Analysis

The task of TTA is to transform a terrain map into a traversability map by assigning a TI value to each cell in the terrain map. This process is divided into two steps: estimating terrain slopes and roughness, and computing the TI value for each cell.

A square terrain patch $P$ centering at the Robot Geometric Center (RGC) is formed in the terrain map. In this paper, $P$ is chosen in such a way that it exactly envelops the robot regardless of the robot’s orientation. The number of data points in the patch is $N = (2L+1) \times (2L+1)$, where $2L+1$ is the number of cells each side length of $P$ has. We then fit a least-square plane to $P$. The normal to the least-square plane $n = (n_x, n_y, n_z)$ and the residual of the fit $\sigma$ are found by using the Singular Value Decomposition (SVD) method [13].

5.1 Schem 1 for TTA

For this scheme, the slope of the terrain patch is estimated by

$$ \phi = \cos^{-1}(n_z), $$

(8)

The TI value of cell $(i, j)$ is calculated using $\sigma$ and $\phi$ as follows:

$$ \tau_{i,j} = F_1 |\phi| + F_2 \sigma / N $$

(9)

5.2 Schem 2 for TTA

Scheme 1 has been implemented in a Segway Robotic Mobility Platform (SRMP) recently [14]. The disadvantage of using Eq. 9 is that it doesn’t reflect the fact that the level of difficulty for a robot to traverse a slope is related to the robot’s yaw angle. To account for this, an alternative method of computing TI is proposed. As the method computes a TI along a specific direction, the robot yaw angle $\psi$ is known. Considering that the robot is on the fitted plane whose normal vector is $n$, the roll and pitch of the robot can be estimated by:

$$ \phi = \sin^{-1}(n_x \cos \psi + n_y \sin \psi) $$

$$ \theta = \sin^{-1}((n_z \sin \psi - n_y \cos \psi) / \cos \phi) $$

(10)

The TI value of cell $(i, j)$ is then calculated by

$$ \tau_{i,j} = F_1 \max(|\phi|, |\theta|) + F_2 \sigma / N $$

(11)

Since the first term depends upon the yaw angle $\psi$, the use of the TI in Eq. 11 allows the robot to steer into a direction that minimizes $\tau_{i,j}$. This enable the robot to snake up/down a steep ramp. $F_1$ and $F_2$ in Eq. 9 and Eq. 11 are chosen by simulation runs in typical urban environments. In this paper, $F_1=3.7$ and $F_2=0.15$ are used and this set of parameters results in the roughness having a larger contribution than the slope.

Apparently, the TTA process expands an obstacle’s boundaries in order to account for the dimensions of the robot. This allows us to treat the robot as a single point in planning the robot motion.

6. The PTI Histogram for Path Planning

The path planning algorithm determines the robot’s next heading direction and velocity based on the traversability property of the local terrain map surrounding the robot. In this work, the traversal property is described by the form of Polar Traversability Index (PTI).

In order to compute PTIs, a square-shaped local traversability map $S^*$ obtained by the above TTA method is formed at the RGC. There are $w_x \times w_y$ cells in $S^*$ and the RGC is at the center of $S^*$. Figure 4 is the traversability map obtained by applying the TTA method to a computer generated elevation map. A white grid in Fig. 4 represents a cell with zero TI while the other grids represent cells with nonzero TIs. A cell with nonzero TI in the traversability map generates an imaginary vector field, which exerts a

![Fig. 4. Transformation of a traversability map into a histogram. $X_0$ and $Y_0$ axes represent the world coordinates; $o$ is the RGC. For simplicity, each sector is drawn in $10^\circ$ and only half of the sectors are shown.](image)

virtual repulsive force on the robot and pushes it away from the cell. This is called the “traversability field” that is defined in terms of a direction and a magnitude computed in the same way as in [15].

The direction of the traversability field generated by cell \((i, j)\) is given by

\[
\varepsilon_{i,j} = \tan^{-1}\frac{y_j - y_0}{x_j - x_0}
\]

and the magnitude of the traversability field is

\[
m_{i,j} = \tau_{i,j}(d_{\text{max}} - d_{i,j}),
\]

where \(x_i, y_j\) are the coordinates of cell \((i, j)\); \(x_0\) and \(y_0\) are the present coordinates of the RGC; \(\tau_{i,j}\) is the TI value of cell \((i, j)\); \(d_{\text{max}}\) is the distance between the vertices (the four farthest cells) of \(S^*\) and the RGC; and \(d_{i,j}\) is the distance between cell \((i, j)\) and the RGC. In Eq. 13, \(\tau_{i,j}\) is squared such that the impact of a cell with a small/large TI value is diminished/magnified. Obviously, \(m_{i,j}\) decreases with increasing \(d_{i,j}\) and it is zero for each of the four vertices of \(S^*\).

In order to evaluate the overall difficulty traversing along a direction, the magnitude of the traversability field produced by all cells in the same direction should be summed up. For this reason, \(S^*\) is divided into \(n\) sectors, each of which has an angular resolution of \(\zeta=360^\circ/n\). Each sector \(k\), for \(k = 0, \ldots, n-1\), has a discrete angle \(\rho = k\zeta\). Cell \((i, j)\) in \(S^*\) is assigned to the \(k\)th sector according to

\[
k = \text{int}(\varepsilon_{i,j}/\zeta)
\]

For sector \(k\), the magnitude of the traversability field produced by all cells in this sector is calculated by

\[
h_k = \sum_{i,j} m_{i,j}
\]

\(h_k\) is the distance-weighted sum of the squared TIs of all cells in sector \(k\). Apparently, a larger value of \(h_k\) means that direction \(\rho = k\zeta\) is harder for the robot to move along with. In other words, \(h_k\) is a TI representing the overall difficulty of traversing the terrain in the corresponding direction. Therefore, it is called the Polar Traversability Index (PTI).

In this paper \(\zeta=5^\circ\), i.e., there are 72 sectors. The PTIs are represented in a form of histogram (see Fig. 5). The path planning method then clusters the histogram into candidate valleys (comprising consecutive sectors with PTIs below a threshold) and hills (comprising consecutive sectors with PTIs above the threshold). One of the candidate valleys that is the closest to the target direction and incurs the smallest heading change is selected as the winning valley. Sector \(k_h\) located inside the winning valley is then chosen as the robot’s next heading direction. The details on how the path planning method determines the robot’s next heading direction are referred to [14].

Since the controller of the robot accepts steering rate and velocity as the control commands, the robot’s next heading direction is used to compute the steering rate as follow:

\[
\hat{\psi} = (k_h(t+1) - k_h(t))\frac{\xi}{\Delta T},
\]

where \(k_h(t)\) and \(k_h(t+1)\) are the robot’s heading directions at time steps \(t\) and \(t+1\), respectively; and \(\Delta T=93.3\) ms is the control interval that equals to the time for the navigation system to acquire 7 sets of laser data. When the robot heads on an obstacle, a reduction in velocity is required since this gives the robot more time for ON. It also allows the robot to obtain more range data on the potential hazard. For this reason, the robot velocity is calculated by

\[
v = v_{\text{max}}(1 - \min(h_m, h_w)/h_m)
\]

where \(v_{\text{max}}\) is the robot’s maximum velocity, \(h_m\) is the PTI in the robot’s current heading direction, and \(h_w\) is a constant that is empirically chosen to produce a sufficient reduction in velocity. A reduction in velocity is also required when the robot approaches the target. In this work, the velocity is further reduced by

\[
v = v \min(d, d_m)/d_m
\]

where \(d\) is the distance between the target and the RGC, and \(d_m\) is a constant (1.5 m in this paper). Note that \(\hat{\psi}\) and \(\psi\) are actually the projection of the steering rate \(\dot{\psi}\) and velocity \(V\) on \(x, y\) plane. They are converted to \(\dot{\psi}\) and \(\psi\). A low pass filter is used to smooth the steering and velocity commands before being sent to the motion controller. In the system, the smoothed commands is computed by

\[
\Theta_i(t+1) = \lambda \Theta_i(t) + (1 - \lambda) \Theta_i(t+1)
\]

where \(\Theta_i(t) = (\dot{V}_i(t), \psi_i(t))^T\) and \(\Theta_i(t) = (\dot{V}_i(t), \psi_i(t))^T\). \(V_i(t)\) and \(\psi_i(t)\) are the smoothed velocity and steering commands that are actually applied to the robot. A larger value of \(\lambda\) produces smoother control commands but a slower system response. \(\lambda=0.75\) is used in the system to provide a relatively smooth control.

7. SIMULATION AND EXPERIMENTAL RESULTS

7.1 Simulation Results

The efficacy of the PTI is first tested on simulated urban terrain. A simulator capable of generating arbitrary urban terrain was developed for the simulation study. The simulator simultaneously renders the animation of the robot traversing the terrain and the PTI histograms in real-time. Figure 5 shows one of the runs over simulated urban terrains. The run started at \(S_1\), terminated at target \(G_1\). Fig. 5a depicts the simulated urban terrain which contains curbs (height=12 cm) and two ramps. The robot moved straightly toward the target in segments \(S_1 \rightarrow 1, 3 \rightarrow 4\) and \(6 \rightarrow G_1\). At point 1 and 4, the curb created a peak in the histogram (Fig. 5b depicts the histogram at point 1) and this initiated the obstacle avoidance from 1 to 2 and from 4 to 5. At point 2 and 5, the
ramp generated a winning valley in the histogram (Fig. 5c depicts the winning valley V2 at point 2). The path planning algorithm started aligning the robot’s heading direction with the ramp and guided the robot passing through the ramp. Eventually, the robot came to a full stop at target G1.

Figure 6 depicts a scenario with the robot climbing up a ramp. The ramp was too steep for the robot to move up straightly if the existing TI measure or Eq. 9 is used. The proposed PTI allowed the robot to snake through the ramp. At each point of S2, 2, and 4, the PTI histogram has a winning valley along the direction with a yaw angle of 50°. (The target direction is with a yaw angle of 90° and it is not traversable due to the large slope angle.) Therefore, the robot moved up the ramp with a 50° heading direction until it reached 1 and 3 where the right curb generated PTI histogram hills and blocked the robot from moving further. The robot then turned to a direction with a yaw angle of 140°. In this way, the robot snaked up the ramp to its target T2.

7.2 Experiments with the Pioneer Robot

The complete ON system was tested with the pioneer robot on an outdoor terrain outside the Fribourgh Hall at the University of Arkansas at Little Rock. As shown in Fig. 7, the terrain is uneven and contains curbs and a ramp. A number of cardboard boxes were used to construct obstacle courses. A navigation task was specified to the robot by giving the robot its target location. The ON system was initialized using the robot’s current position as its start point (the robot X, Y, Z coordinates are reset to zero).

The ON system was implemented on the onboard computer that runs RTLinux as the Operating System. The LRF’s angular resolution was set to 1°. Each scan of the LRF takes 13.3 ms and contains 181 range data that covers a 180° field of view. The Real-time Data Acquisition (RDA) section in the ON system (Fig. 2) resides in the real-time Linux kernel as a real-time thread. The RDA thread buffers the LRF data, the gyroscopic data, and the wheel encoder data in a shared memory. Once it obtains 7 scans of LRF data and the associated gyroscopic and encoder data, the Map-building and Filtering section fetches this data set; and registered the range data into a terrain map. The TTA module then obtains the local terrain map (55×55 cells) and transforms it into a traversability map. Then the path planning module converts the traversability map into a PTI histogram, determines the motion commands, and sends the commands to the robot. The above process repeats every 93.3 ms (7×13.33 ms) until the robot reaches the target.

The robot pose information (essential to terrain mapping and target seeking) were obtained by a sensor fusion method that estimate the robot roll, pitch and yaw angles, and X, Y, Z coordinates based on the encoder data and gyroscopic data (the Euler rate). The sensor fusion method is omitted here due to space limit.

Figure 7 shows one of the autonomous runs from S1 to G1 on a simulated urban terrain. Kt–target direction, Kh–robot heading direction, L/R–left/right border of a candidate valley, t–time step, Vel–velocity; Yaw–robot yaw angle.
in black and white, a brighter point has a bigger elevation value.) Figure 7c displays the elevation map at the point where the robot was avoiding the first obstacle it encountered. Figure 7e shows the elevation map at the time when the robot was avoiding the curb and aligning itself with the ramp. Figure 7f shows the elevation map after the robot successfully traversed the ramp. In the elevation maps, the unperceived areas (e.g., area c) of the lower terrain surface look like obstacles because their Z coordinate values (zero) are bigger than that of the surrounding terrain surfaces whose elevation values are negative.

Data collected from all experimental runs has demonstrated that the roll, pitch, and yaw angles, and the velocity of the robot changed smoothly. This means that the navigation method produced smooth motion commands for the robot.

Figure 8 depicts the robot’s roll, pitch and yaw angles during the experimental run from $S_3$ to $G_3$ in Fig. 7. The roll and pitch angles are limited in $(-1^\circ, 8^\circ)$ and $(-3^\circ, 7^\circ)$, respectively. This means that the navigator guided the robot to traverse moderate terrain. Both roll and pitch angles changed smoothly during the entire navigation task except for the points around ‘S’ in Fig. 8a and Fig. 8b where relatively larger changes in roll and pitch are observed. This is because the robot ran over the left curb at the upper end of the ramp (labeled by $A$ in Fig. 7a). The curb is low-profiled and considered traversable. However, it generated relatively larger changes in the robot’s roll and pitch angles. As shown in Fig. 8c, the robot’s yaw angle changed very smoothly during the whole process. This indicates that the navigation method generated smooth steering commands for the navigation of the robot.
8. CONCLUSIONS

In conclusion, this paper introduces a terrain mapping system and a terrain traversability analysis method for mobile robot navigation in urban environments. The terrain mapping system uses a 2-D LRF and is thus affordable. The advantage of the proposed PTI over the conventional TI is that it allows a robot to analyze terrain traversability along each direction, and thus enhances the robot’s obstacle negotiation ability, i.e., enables the robot to snake through a steep ramp which may be untraversable if the conventional TI is used. The PTI concept and the terrain mapping method have been validated in a complete navigation system on a real mobile robot platform in outdoor urban environments. The entire system can be ported to any other mobile robot for terrain mapping and navigation as it only requires wheel encoder data from the robot. The system can be applied to safe navigation of mobile robots/vehicles in urban environments for transport, surveillance, patrol, search and rescue, and military missions.

References


Cang Ye received B.E. and M.E. degree from the University of Science and Technology of China, P. R. China in 1988 and 1991, respectively. He received his Ph.D. degree from the University of Hong Kong in 1999.

He is currently an Assistant Professor in Department of Applied Science, University of Arkansas at Little Rock, AR. He was a research investigator at University of Michigan, Ann Arbor, MI from 2003 to 2005. His research interests are in mobile robot terrain mapping and navigation, fuzzy systems and reinforcement learning.

Dr. Ye is a senior member of IEEE and a member of Technical Committee on Robotics and Manufacturing Automation, IEEE Systems, Man, and Cybernetics Society.

Wen-Fang Xie is an assistant professor with Department of Mechanical and Industrial Engineering at Concordia University, Montreal, Canada. She was in Industrial Research Fellowship holder from Natural Sciences and Engineering Research Council of Canada (NSERC) and served as a senior research engineer in InCoreTec, Inc. Canada before she joined Concordia University in 2003.

She received her Ph.D from The Hong Kong Polytechnic University in 1999 and Master Degree from Beijing University of Aeronautics and Astronautics in 1991. Her research interests include nonlinear control in Mechatronics and Robotics, Neural Network, system identification and advanced process control.