A MULTI-SENSOR-BASED NAVIGATION FRAMEWORK FOR INTELLIGENT VEHICLE

Photchara Ratsamee, Surapa Thiemjarus* and Toshiaki Kondo

Received: Sept 15, 2010; Revised: Nov 7, 2010; Accepted: Nov 9, 2010

Abstract

This paper presents a platform-independent framework for autonomous navigation of an intelligent vehicle. The framework consists of three integrated modules, namely; waypoint navigation, obstacle localization and path planning. Each module has been individually validated based on experiments with a real intelligent vehicle. For waypoint navigation, we propose the use of Google Earth for generation of reference waypoints and a simple bias subtraction method for GPS calibration. Based on the autonomous navigation experiments, this method yields a more stable navigation path compared to the use of GPS-generated waypoints and translational error can be efficiently eliminated. For obstacle localization, we develop a feature-based approach for obstacle detection and map generation based on the use of compactness measure and perspective projections. With an integrated use of a camera, digital compass, and GPS, static obstacles of a known dimension, along with their positions and orientations on the road can be calculated in real-time while the vehicle is travelling. Based on the derived information, an overhead-view obstacle map is generated to provide an internal representation of the road. The experiment on an unmarked road shows that the estimation of an obstacle can be achieved with maximum errors of 1.4 degree, 15 cm and 12 cm in angle, depth and lateral positions, respectively. Based on the obstacle map and the Google Earth waypoints, artificial potential field is adopted for collision-free path generation.

Keywords: Intelligent vehicle, waypoint navigation, google earth, GPS, obstacle detection, obstacle map generation, path planning

Introduction

Nowadays, there exist over 800 million vehicles on the road worldwide and the number is projected to grow over 1.2 billion by 2020 (Webber, 2005). This increasingly accumulated number has raised several issues concerning safety of road users and pedestrians. To improve driving safety as well as to provide a more convenient and efficient support to the

School of Information, Computer, and Communication Technology. Sirindhorn International Institute of Technology, Thammasat University, Pathumthani, 12121, Thailand. Tel: 0-2501-3505-20 ext. 1813, Fax: 0-2501-3524, E-mail: photchara.ratsamee@gmail.com, {surapa, kondo}@siit.tu.ac.th

* *Corresponding author*

drivers, the concept of intelligent vehicle was proposed (Richard, 2005; Siciliano and Khatib, 2008). Intelligent vehicle is an autonomous robot that can move according to predefined waypoints without a driver or remote control. To further facilitate its user, some tasks that humans perform while driving a vehicle can also be automated.

Navigation system is a fundamental building block of an unmanned intelligent vehicle system. Recently, most advanced systems are navigated based on a predefined set of reference waypoints and a Global Positioning System (GPS). Waypoints are basically locations in a physical space that are pre-stored in memory so that the same path can be traced at a later time. GPS, on the other hand, is a satellite-based system that can provide the latitude and longitude information of the receiver by calculating the time difference of signals traveling from different satellites to the receiver (Hofmann-Wellenhof *et al*., 1993). It is a popular technique for locating the current position of the vehicle. By iterative comparison of the current coordinates obtained from a GPS while traveling with reference waypoints, the position of the vehicle can be adjusted to move along the pre-defined path. Since the accuracy of GPS can significantly affect the reliability of a navigation system, many approaches for improving GPS accuracy have been implemented (Sukkarieh *et al*., 1999; Xiangdong *et al*., 2001; Cui and Ge, 2003; Naranjo, 2004; Hayashi *et al*., 2008; Limsoonthrakul *et al*., 2009).

Several prototypes of an intelligent vehicle have been developed by the research community. Stanley (Thrun *et al*., 2006), for example, is a famous intelligent vehicle developed by the Stanford racing team. It is a high-speed desert driving vehicle which can perform localization through a probabilistic reasoning based on the information from heterogeneous sensors, such as LIDAR (Light Detection and Ranging), GPS and a camera. Another famous example is Skynet (Miller *et al*., 2008), developed by Cornell. Its reasoning engine consists of many subsystems such as a vision-based obstacle detection module, an optimization-based path planner, and a state-based reasoning agent which adaptively adjust the path according to traffic laws. Sukkarieh *et al*. (1999) developed a navigation system for autonomous land vehicle applications based on the integrated use of GPS and an Inertial Measurement Unit (IMU). Limsoonthrakul *et al*. (2009) performed localization by fusing data from a GPS, a digital compass, a camera and an encoder. Particle filtering was used for sequential state estimation.

The intelligent vehicles used in this study were first developed for participation in the Thailand Intelligent Vehicle Challenge (2007). Based on pre-defined waypoints, the intelligent vehicles are programmed to travel along the test track, on which obstacles with predefined shapes and colors are located. After the competition, we aim towards developing an intelligent vehicle that can be used in real-world applications. Figure 1 shows the two intelligent vehicles used in this

Figure 1. The SIIT and AIT intelligent vehicles

study. The SIIT intelligent vehicle (left) was built based on a Sanyo 3-wheel electric car, and the AIT intelligent vehicle (right) was built based on a golf car. Table 1 provides summarized specifications of the two vehicles.

Both vehicles make use of the integrated information from a GPS, a digital compass, and a camera. The GPS is used for latitude and longitude data acquisition. The magnetic compass measures the average heading angle of the vehicle with respect to the North Pole. The web camera, installed at the front of the steering wheel, is used for obstacle detection. The controlling module consists of a driving motor and a steering motor, powered by lead-acid batteries. A laptop is used as the central processing unit.

This paper presents a framework for collision-free navigation along a predefined path with relatively constrained obstacle positions. The framework consists of three modules, i.e., 1) Google Earth-based waypoint navigation, 2) feature-based obstacle localization and 3) collision-free path planning with potential field (Passino, 2004). The obstacle map provides an internal representation of the world and is used for offline path planning. This can be combined with reactive obstacle avoidance basing on local information and the current state of the perceptual input for handling unexpected events. The schematic diagram of the proposed sensor fusion architecture is illustrated in Figure 2.

Google Earth-Based Waypoint Navigation

Navigation System

A predefined path and the position of the vehicle are two important pieces of information required by an autonomous navigation system. Both waypoint and current vehicle's coordinates consist of latitude, e , and longitude, n , components. In a navigation system, i^{th} the waypoint, w (i), is defined as:

$$W(i) = [e_w(i), n_w(i)] \quad (1)$$

The position of the vehicle at time t, V (t), is defined as:

$$V(t) = [e_v(t), n_v(t)] \quad (2)$$

Assuming the vehicle is travelling at a constant speed, moving the vehicle to a specified position can be achieved via controlling of the steering motor (*i.e.* local heading direction of the vehicle). The steering angle, γ, is derived based on the current global position of the vehicle and the two consecutive waypoints using (3) to (5). The graphical illustration of parameters involved the calculation is depicted in Figure 3.

The angle between the two consecutive waypoints with respect to the East axis, θ_w , is calculated using:

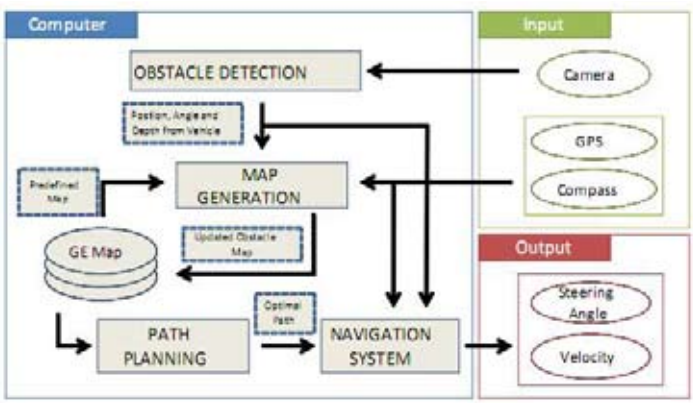


Figure 2. Overall system architecture of the proposed navigation framework

$$\theta_w = \tan^{-1}\left(\frac{(n_w(i) - n_w(i-1)) \times K_{NS}}{(e_w(i) - e_w(i-1)) \times K_{EW}}\right) \quad (3)$$

where K_{EW} = 106,080 meters/degree and K_{NS} = 109,369.2 meters/degree are constant parameters for converting the geographic coordinates (latitude/longitude) into the metric coordinates.

The angle of the vehicle with respect to the next waypoint, θ_v, is calculated by

$$\theta_v = \tan^{-1}\left(\frac{(n_w(i) - n_v(t)) \times K_{NS}}{(e_w(i) - e_v(t)) \times K_{EW}}\right) \quad (4)$$

Finally, the steering , γ , which is used to control the steering wheel of the vehicle, is defined as:

$$\gamma = \tan^{-1}\left(\frac{-\cot(\theta_v)}{K}\right) - \theta_w - \beta \quad (5)$$

where $0 < K < 1$ is the tracking constant and β is the current heading direction of the vehicle obtained from the digital compass. The value of K depends on the size of the car and is set to 0.5 for AIT and 0.8 for SIIT intelligent vehicles to ensure the smoothness of the path. More detailed derivation of these equations can be found in (Ratsamee *et al.*, 2010).

Waypoint Generation and Calibration

A popular technique for determining the current vehicle's position is by using a GPS. Pre-collecting GPS data is also a common method for waypoint construction. The path can be created by driving the vehicle to collect GPS waypoints along the test track. An alternative method is to extract the waypoint coordinates from a pre-defined map or a satellite image. In this study, Google Earth (Google Inc, 2009) is used for waypoint generation. A set of waypoints is generated simply by carefully drawing the path on the satellite image (Deelertpaiboon and Parnichkun, 2008).

Assume the GPS value at the starting point, V_0^{GPS}, is reasonably reliable, simple bias subtraction can be used to eliminate the translational errors. Let W_0^{GE} be the 2D coordinate of the starting position obtained from Google Earth (GE). An offset vector, ΔW, is defined as:

$$\Delta W = W_0^{GE} - V_0^{GPS} \quad (6)$$

The vehicle's position along the path is then translated onto the GE coordinate as follows:

$$V_t^{GE} = V_t^{GPS} - \Delta W \quad (7)$$

Feature-Based Obstacle Localization

This section presents a method for obstacle detection and the obstacle map generation based on a passive monocular color camera, digital compass and GPS. The proposed method consists of two main steps, *i.e.*, 1) obstacle detection and 2) estimation of the locations and orientations.

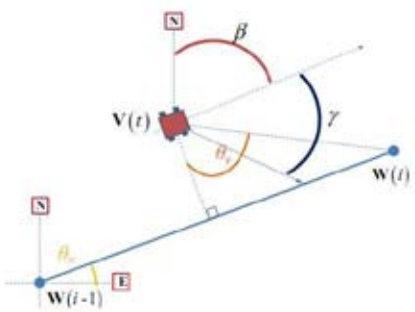


Figure 3. Parameters involved in the steering angle computation

Obstacle Detection

To differentiate an obstacle from the background scene, information regarding objects' properties such as color, shape and geometrical parts can be exploited. We first use color to segment obstacles from the background scene. Thresholding is applied on the hue and saturation components in the HSV color space to determine whether each pixel belongs to an obstacle. Unlike RGB, HSV represents intensity or brightness in a dimension orthogonal to color and thus is more robust in low light intensity conditions (Gonzalez and Richard, 1992). To smoothen the resulting image, morphological opening (Vincent, 1992) is applied for noise removal. Detected pixels are represented as white regions, known as blobs, from which properties such as perimeter, area and centroid can be extracted for further analysis.

Obstacle's shape is another useful piece of information that can be used to enhance segmentation accuracy. As an indicator of the object's shape, the classical compactness measure (Ballard and Brown, 1982), *i.e.*,

$$Compactness = \frac{Perimeter^2}{4\pi \times Area} \qquad (8)$$

which is independent of geometrical transformations such as translation, rotation and scaling (Bribiesca, 1997), is used to filter out regions that are unlikely to be the object of interest.

For more accurate blob extraction, further analysis can be made using the geometrical information of the obstacle. In this study, the obstacles are square-shaped. Their edges are extracted using Hough transform and corners are detected from the intersection of the Hough lines.

Estimation of Obstacles' Locations and Orientations

The next step of obstacle map generation is to estimate the (top view) position, $0 = (x_O, z_O)$, and orientation with respect to the North Pole, θ_N, of each obstacle. Based on these two parameters, the obstacles can be located on a pre-defined map. In this study, a method based on perspective projection (Carlbom and Paciorek, 1978) is adopted. The 3D perspective model illustrated in Figure 4 describes the relationship between the camera image and the actual object.

The camera location is defined as the vehicle location, $V = (x_v, y_v, z_v)$. The obstacle,

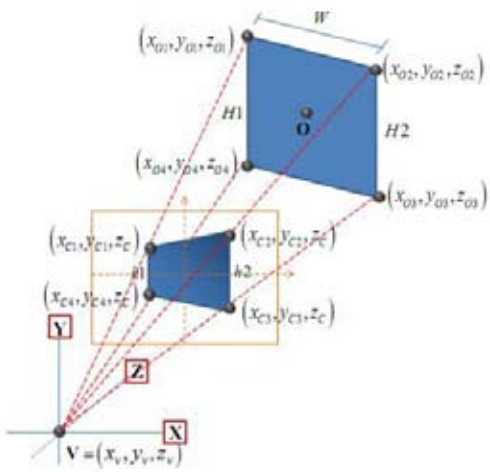


Figure 4. An obstacle in the 3D perpective model

represented as a plane in 3D space, consists of four corners, *i.e.* (x_{O1}, y_{O1}, z_{O1}), (x_{O2}, y_{O2}, z_{O2}), (x_{O3}, y_{O3}, z_{O3}), and (x_{O4}, y_{O4}, z_{O4}), which correspond to the projected feature points (x_{c1}, y_{c1}, z_c), (x_{c2}, y_{c2}, z_c), (x_{c3}, y_{c3}, z_c), and (x_{c4}, y_{c4}, z_c) on the image plane, respectively. The focal length of the camera (scaled to pixel unit), f, is first determined based on the relationship between the measured real obstacle's dimension and location and the projected image pixels as described by the following equation:

$$\frac{f}{D_{\perp}} = \frac{h}{H} \tag{9}$$

where $D_{\perp}$ is the distance of the obstacle from the vehicle along the heading direction, and H are h the vertical heights of the real obstacle (in meters) and the obstacle in the image (in pixels), respectively. Since f and H are known, the above equation can be used to estimate $D_{\perp}$ at run-time based on h observed from the image.

The perspective projection of the obstacle on the image plane depends on the deviated angle or the feature point from the heading direction, $\varnothing$, described as:

$$\varnothing = \tan^{-1}\left(\frac{n}{f}\right) \tag{10}$$

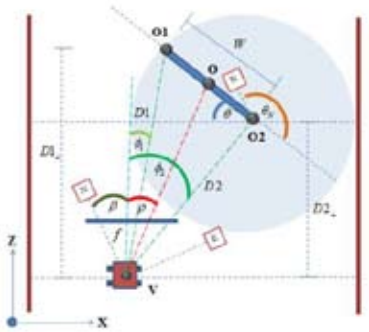


Figure 5. Top view of the vehicle and the obstacle

where n is the lateral distance (in pixels) from the center to the edges of the obstacle in the image and $z_C = f$. Assuming that the obstacle is orthogonal to the xz-plane, the top view of the described 3D perspective model is shown in Figure 5. O1 and O2 denote the obstacle edges after 2D projection onto the xz-plane. $D1_{\perp}$ and $D2_{\perp}$ are distances of the points O1 and O2 from the vehicle, respectively. The perpendicular distances, i.e. $D1_{\perp}$ and $D2_{\perp}$, and the deviated angles of two obstacle edges from the heading direction of the vehicle, *i.e.* $\varnothing_1$ and $\varnothing_2$, can be estimated using (9) and (10). From the deviated angle from the heading direction and the position of vertical edge, O1 = (x_{O1}, z_{O1}) is calculated using:

$$x_{O1} = x_v + D1_{\perp} \times \tan\left(\phi_1\right) \tag{11}$$

$$z_{O1} = z_v + D1_{\perp} \tag{12}$$

Similarly, the position of the other vertical edge O2 = (x_{O2}, z_{O2}) is described as:

$$x_{O2} = x_v + D2_{\perp} \times \tan\left(\phi_2\right) \tag{13}$$

$$z_{O2} = z_v + D2_{\perp} \tag{14}$$

The width of the obstacle can be estimated as follows:

$$W = |\mathbf{O1} - \mathbf{O2}| \tag{15}$$

This value can be compared to the actual width of the obstacle in order to verify the distance estimation algorithm.

The heading direction of the vehicle obtained from the digital compass, β, is used to create the reference line:

$$L_v = \beta + \frac{\pi}{2} \tag{16}$$

With the known obstacle's width, W, the turning angle, θ, between the vehicle reference line, L_v, and the obstacle reference line, L_0, is determined as:

$$\theta = \sin^{-1}\left(\frac{\left|D1_{\perp} - D2_{\perp}\right|}{W}\right) \quad (17)$$

To generate obstacle on the map, the turning angle compared to the North Pole, θ_N, and the position of the obstacle, O = (x_o , z_o), are calculated as follows:

$$\theta_N = \theta + L_V \quad (18)$$

$$\mathbf{O} = \frac{\mathbf{O1} + \mathbf{O2}}{2} \quad (19)$$

From the estimated position of obstacle, the angle between the vehicle and the obstacle plane, ρ , is derived as follows:

$$\rho = \tan^{-1}\left(\frac{x_O}{z_O}\right) \quad (20)$$

The feature-based obstacle detection can be activated when a specific condition is met or iteratively computed at a regular time interval. The estimated parameters are stored for further use in path planning.

Collision Free Waypoint Generation

In the proposed navigation framework, reference waypoints of a predefined path and the dimension of the road are first extracted from Google Earth map. To avoid the obstacles, the vehicle has to modify the trajectory with the obstacle information obtained from the camera system using the method described in Section 3. For real-time collision-free navigation, the artificial potential field method is used. To avoid the obstacles while remaining within the road boundary, an intelligent vehicle travels from an initial position to the target position (goal) based on the concept of attractive and repulsive forces.

The attractive internal force, F_g (P), from the goal position, P_g , to an arbitrary point , P = (x , z), is defined as

$$F_g\left(\mathbf{P}\right) = \omega_g \times \left(\mathbf{P} - \mathbf{P}_g\right)^2 \quad (21)$$

where ω_g is the weight of the attractive force from the goal position, P_g. This force keeps the vehicle's trajectory towards the target. At every position of the vehicle, the internal force vector usually directs towards the goal position.

The external force is a repulsive force originated from the road boundary, P_b , or an obstacle, P_O , as follows:

$$F_b\left(\mathbf{P}\right) = \omega_b \times \exp\left(k_1 \times \left(\mathbf{P} - \mathbf{P}_b\right)^2\right) \quad (22)$$

$$F_o\left(\mathbf{P}\right) = \omega_o \times \exp\left(k_2 \times \left(\mathbf{P} - \mathbf{P}_o\right)^2\right) \quad (23)$$

where ω_b and ω_o are the weights of the repulsive force from P_b and P_O , respectively. and k_1 are k_2 the influence constants which protect the vehicle from collision. When an obstacle is detected, a fixed window of dimension $l_1 \times l_2$ is located at the obstacle's center. This window indicates an active region over which the repulsive force field is generated from the obstacle.

The navigation path to the target can be obtained by tracing through waypoints with locally minimum resultant force defining as a combination of the three force fields:

$$R\left(\mathbf{P}\right) = F_b\left(\mathbf{P}\right) + F_o\left(\mathbf{P}\right) + F_g\left(\mathbf{P}\right) \quad (24)$$

At each time step, a possible waypoint, P_s (i), that the vehicle can move to is

$$\mathbf{P}_s\left(i\right) = \mathbf{P} + r \times \left[\cos\left(\frac{2\pi i}{N}\right), \sin\left(\frac{2\pi i}{N}\right)\right] \quad (25)$$

where $0 \le i \le N$ denote the direction number, r is the sensing radius and N is the total number of possible directions around **P**. The next waypoint, **P**´, is defined as

$$\mathbf{P}' = \min_{0 \le i \le N}\left(R\left(\mathbf{P}_s\left(i\right)\right)\right) \quad (26)$$

Experiment and Results

The experiments in Sections 5.1 and 5.2 were conducted on an obstacle-free road in a car

park inside SIIT Rangsit campus. The road has smooth surface and a distance of ~500 meters. Figure 6 illustrates the experimental venue, along with the path generated using Google Earth. In Section 5.1, our initial investigation on the characteristics of GPS errors will be first discussed. Section 5.2 will then demonstrate the results on self-navigation of the SIIT intelligent vehicle using the GE-based waypoint correction. Section 5.3 illustrates the experiment on feature-based obstacle localization with the use of monocular camera. Section 5.4 presents the collision-free waypoints generated using the potential field method described in Section 4. The last two experiments were conducted on an unmarked road inside AIT using the AIT intelligent vehicle

An Investigation of GPS Errors

At a sampling rate of 4 Hz, GPS datasets were collected by manually driving the car along the test track in the morning, afternoon, evening and on a cloudy day, respectively. A plot of the GPS data acquired along the path under different weather conditions is shown in Figure 7. This indicates that the reliability of the GPS data is highly weather-dependent and if the GPS-based waypoint construction is conducted on a cloudy day, the intelligent vehicle will run out of the road.

Two sets of reference waypoints were acquired, one from Google Earth and the other from the GPS while driving the vehicle along the test track. As shown in Figure 8, there exists a reasonable amount of misalignment between the path generated using the GPS data and the path generated using Google Earth. Most of which are translational errors. Consequently, the waypoints generated by Google Earth cannot be directly used by the vehicle.

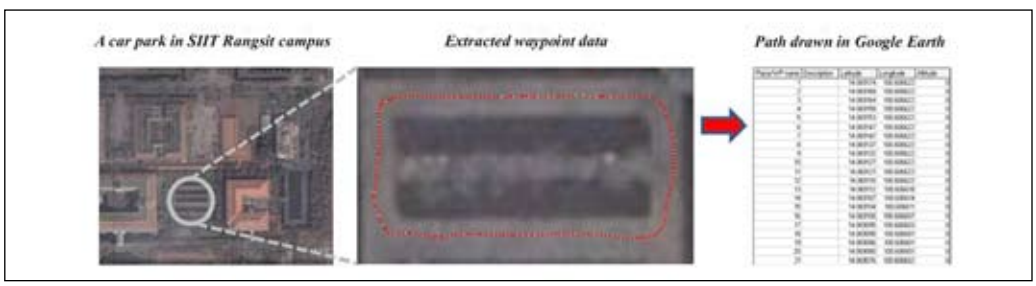


Figure 6. Plots of the GPS data acquired while driving the car on the test track at different time and weather conditions

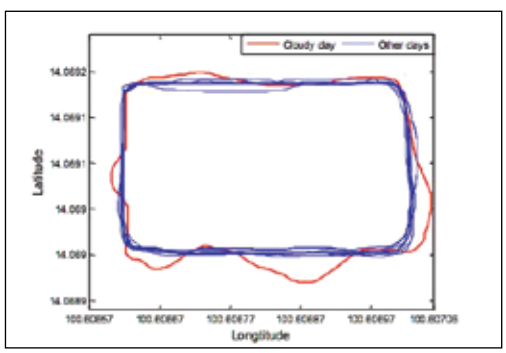


Figure 7. Path generation using the GE-Path program

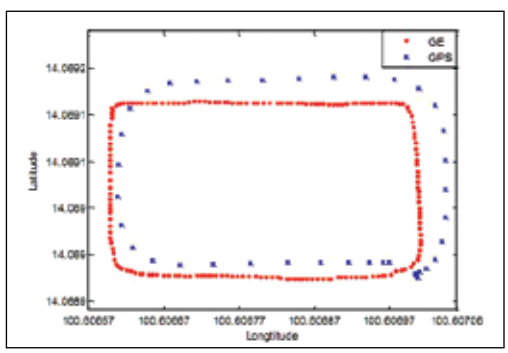


Figure 8. Plots of waypoints generated by using Google Earth and the GPS data

Google Earth Waypoint Correction

To eliminate the translational errors, the GPS correction algorithm described in Section 2.2 was applied. Six iterations of self-navigation were performed by the SIIT intelligent vehicle. One of which is based on the pre-collected waypoints using GPS and the rest are based on the corrected reference waypoints obtained from the proposed algorithm. Figure 9 demonstrates the plots of GPS data acquired while the vehicle was travelling along the test track using the two waypoint generation schemes. From the results, it is obvious that the GPS-based navigation deviates significantly from the reference path as the effect of GPS errors can accumulate through time during the GPS waypoint navigation. With the GE-corrected waypoints, the errors between each step are more or less independent. Table 2 summarizes the total errors calculated for the six self-navigation experiments. In average, the total error between the self-navigated path using Google Earth waypoints and the corrected GE reference path is ~1.3 m With the GPS-based waypoints, the total error increases up to 7.4 m

Obstacle Localization and Map Generation

To obtain a collision-free path, accurate estimation of obstacles' locations and orientations on the road is an important part of both offline and online path planning. As an initial investigation, we simplified the problem by using rigid rectangular obstacles with a fixed dimension of 1.5 × 1.5 m. The obstacle plates represent a basic geometrical shape that comprises several real-world objects, thus facilitate the extension of the method to more generic obstacle patterns. Green and red obstacle plates were randomly placed on an unmarked road inside Asian Institute of Technology (AIT) in varying locations and orientations.

Using the starting point of the vehicle as the reference point, the road dimension along with position and orientation of each obstacle were measured and used as the ground truth for result validation. The width of the road varies between 3 to 5 m and the total distance of the path is approximately 200 m. The traveling speed of the vehicle varies between 1 to 2.5 m/sec. During vehicle navigation and obstacle avoidance along the test track, the GPS, compass and image data are collected simultaneously. For GPS and digital compass, the sampling rate of 10 Hz is used. Input images are collected at 20 frames/sec using a low-cost web camera. The image resolution is set to 320 × 240 pixels with 8-bit intensity levels for R, G, and B channels. The camera has a 68 degree horizontal field of view and is placed on the console 0.75 m above the ground.

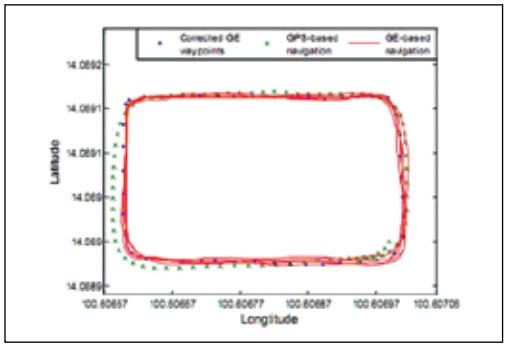


Figure 9. Plots of GPS coordinated of the vehicle the six trials of GPS-based and GE-based self-navigation

Figure 10. An illustration of the test track obtained from Google Earth, obstacle locations and navigation path of the vehicle during one iteration of data collection

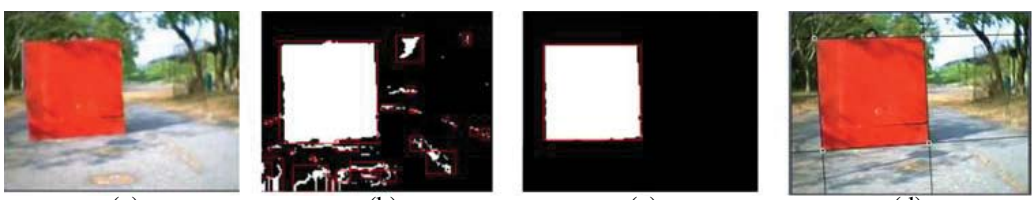
Two rounds of data collection were made with 5 obstacles placed in different positions and orientations for each round. Figure 10 illustrates the locations of the obstacles during one round of data collection.

The dotted line illustrates the navigation path of the vehicle based on GE-corrected waypoints. Examples of different obstacle images captured during vehicle navigation along the path are shown in Figure 11. A calibration experiment is first conducted to estimate the focal length of the camera, f, and the threshold values for compactness, hue, and saturation of the obstacles. The focal length was estimated from the frontal images of obstacles captured at varying distances. The threshold values are experimentally specified based on color and shape of the obstacles. The resulting camera and threshold parameters are described in Table 3. Figure 12 illustrates different steps in extracting obstacle's features from an image. The original RGB image in Figure 12(a) is first converted into the corresponding HSV color space, after which thresholding on hue and saturation is applied, resulting in the black and white image as shown in Figure 12(b). Figure 12(c) shows the result image after morphological opening (with a 4 × 4 squared mask) and thresholding on compactness are applied. Finally, edge detection and Hough transform are then applied and the obstacle features, *i.e.* edges and corners, can be extracted from line intersections, as shown in Figure 12(d). The extracted features are used to determine the height, h, and the deviated angle, $\varnothing$, of the obstacle in the image for further estimation of its location and orientation of the obstacle using the method described in Section 3.2. All data processing is performed in real-time using MATLAB Simulink.

The proposed feature-based obstacle detection is performed iteratively at a constant time interval. In this study, the encoder associated to the steering motor is the module with the slowest update rate of 4 Hz. The update rate

(a) (b) (c) (d)

Figure 11. Example images of obstacles labeled in Figure 10 : (a) 1st obstacle; (b) 2st obstacle; 3st obstacle; and 4st obstacle



(a) (b) (c) (d)

Figure 12. Resultd of different steps during feature-based obstacle detection: (a) original image, (b) blob image after thresholding is applied on hue and saturation, (c) blob image after applying morphological opening and thresholding on compactness, and (d) original obstacle image with edges and corners extracted using Hough transform. Each red square in the blob images indicates the detection of an obstacle

of the obstacle detection module, therefore, is also set to this frequency.

To validate the performance of the proposed method, generated obstacles' locations, $O = (x_o , z_o)$, and orientations, θ_N , are compared against the actual measured values. The average and standard deviation of angle error, $\Delta\theta_N$, depth error, Δz_o , and lateral error, Δx_o , of each obstacle calculated over frames with detected obstacles are summarized in the second main column of Table 4. From the average frame results, the average error is approximately 1.44 degree in angle, 15.6 and 14.95 centimeters in x- and z-directions, respectively. The values of standard deviation indicate significant amount of variation in inaccuracy across different image frames. The depth error, Δz_o , is caused by an inaccurate estimation of the vertical height, h , of the obstacle in the image. This error occurs when the vehicle is not exactly at the same level as the obstacle. One of the obvious causes of angle error, Δq_N , is the depth error, since q_N is consequently derived from (15). Another possible source of angle error is the inaccurate angle of vehicle heading direction, b , obtained from the digital compass.

For autonomous navigation, lateral error, Δx_o , should be seriously concerned because it may cause the vehicle to run out of the road lanes. Based on a close observation, the lateral error, Δx_o , relies significantly on the distance, $D_\perp$ and the angle between the vehicle and the obstacle plane, ρ , derived using (20). The relationship among Δx_O, $D_\perp$, and ρ in the collected dataset is observed and illustrated as an interpolated error surface in Figure 13. It is shown on the 3D plot that higher values of $D_\perp$ and ρ can amplify the lateral error. This is due to the fact that the deviated distance, n , is a result of mapping an obstacle from real world onto the image plane relative to the curvature of the lens. Therefore, there exists some distortion due to lower pixel resolution as the value of n increases. Quantization error is another source of the inaccuracy in the estimated lateral distance. From (10), higher image resolution will increase the resolution of the deviated angle, $\varnothing$, and therefore shall decrease the lateral error. From Figure 13, the minimal lateral error occurs at distances around 4 meters and angle less than 25 degrees. This is used as the condition for best frame selection. From the selected frame, the locations,

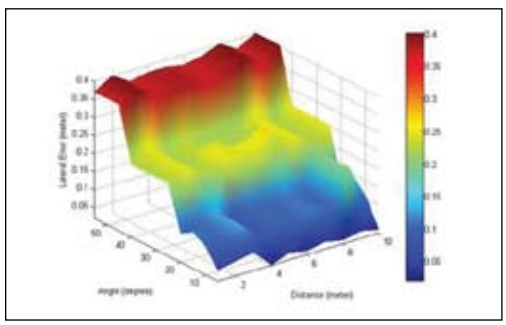


Figure 13. An interpolated surface of the lateral error for varying angles P and distances $D_\perp$

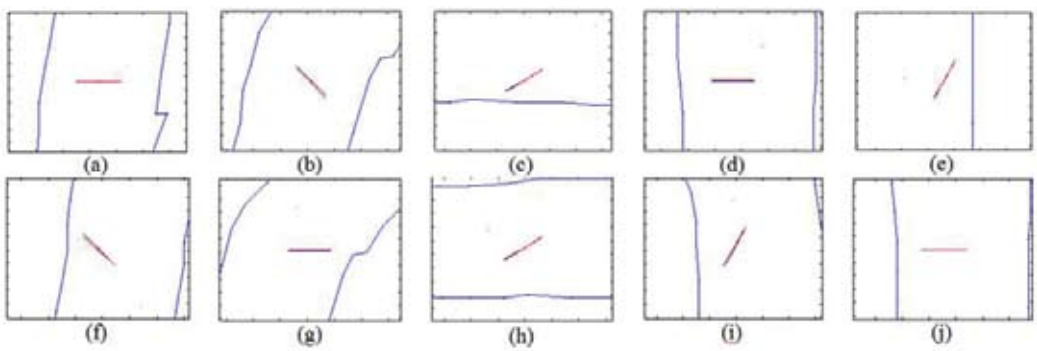


Figure 14. The magnified version of the ten estimated obstacles in the best frame compared with the actual measurement

$O = (x_o, z_o)$, and orientations, θ_N, are estimated for each obstacle. The angle error, $\Delta\theta_N$, depth error, Δz_o, and lateral error, Δx_o, of each obstacle calculated from the selected best frame along with the values of $D_\perp$ and ρ that pass the best frame condition are summarized in the right most main column of Table 4. From the best frame results, the average error is approximately 0.67 degree in angle, 6.8 and 10.3 cen in x- and z-directions, respectively. The estimated and actual locations and orientations of ten obstacles are shown in Figure 14. The blue lines indicate the road boundary generated using Google Earth. The green dotted line is the vehicle path, from which video images are captured. The red and blue line segments are the estimated and the actual locations and orientations of the obstacles, respectively.

Collision-Free Waypoints Generation

To obtain collision-free waypoints, the potential field algorithm described in Section 4 was applied on each detected obstacle. In this study, an active region of size 10 × 15 m was used. Figure 15 illustrates the potential field generation process. Figure 15(a) shows the attractive force field from the target position (the farthest waypoint from the vehicle in the active region). Figures 15(b-c) show the repulsive force fields generated by the boundary and the obstacle, respectively. Figure 15(d) shows the resultant of the three force fields. The collision-free waypoints are obtained by iteratively searching for new minimum points until the final target is reached. The parameters adopted for potential field generation is summarized in Table 5. Figure 16 illustrates the waypoints along the path used for the experiment conducted in Section 5.3, before and after applying the potential field algorithm. The original and the updated waypoints are denoted as blue dots and red crosses, respectively.

The experimental results show that the vehicle always tracks to the final position along a smooth path that passes the wider area of the road due to the low probability of collision. The force field from the road boundary always protects the vehicle from going off the road. This new set of waypoints can be provided

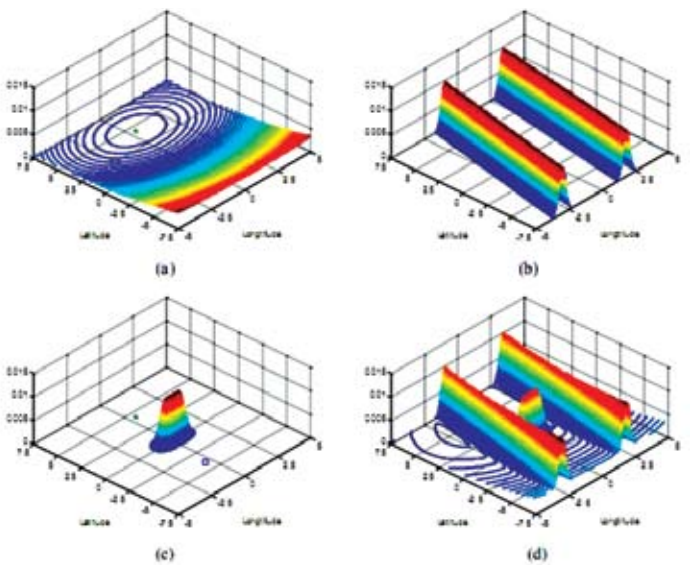


Figure 15. (a) is an attractive forces field generation from start to goal position, repulsive force field generation from (b) considered obstacle and (c) road boundary waypoints and (d) is the resultant force field which is used for obstacle avoidance

directly to an intelligent vehicle for collision-free autonomous navigation. Furthermore, when an unexpected obstacle is detected on the road, this algorithm is capable of generating new waypoints for avoiding the obstacle in real-time.

Conclusions

This study proposed a framework for collision-free waypoint navigation for an intelligent vehicle. A detailed study on GPS errors and a waypoint correction algorithm have been first introduced. It has been demonstrated by several self-navigation experiments that simple bias subtraction can effectively correct and compensate the translational error during travelling. The corrected waypoints obtained from Google Earth provide a high sampling rate and a stable path for waypoint navigation. The technique is simple, practical, and efficient in terms of computational cost and time. A consequence study on feature-based obstacle localization based on the use of a monocular camera has been discussed. Based on the use of compactness and perspective projection, the turning angle and position of a known shape obstacle can be estimated in real-time while the vehicle is travelling. From the experimental results, we found that the effective range that provides the best frame condition of the obstacle for the geometrical computation is ~4 meters, with the angle less than 25 degrees from the heading direction. This is also a suitable condition in which the obstacle avoidance routine is required to be activated during autonomous navigation. The obstacle localization algorithm yields an average error of 0.67 degree in angle, and 6.8 and 10.3 centimeters in x- and z-directions, respectively. The accuracy of the algorithm can be further improved by increasing the image resolution.

The estimated obstacle locations and orientations are processed online while travelling and superimposed with the existing map information. Based on the artificial potential field method, a new set of collision-free waypoints can be obtained. This generated path leads the vehicle to the low probability of collision region. The framework can be also applied in real-time for detecting and avoiding an unexpected obstacle along the path.

For further study, the framework can be further expanded to handle with more generic and complex obstacles. This will improve the practicality and potential usage of the framework towards real-world application scenarios.

Acknowledgments

This research is supported by the Thailand National Electronics and Computer Technology Center (NECTEC) and the SIIT Student Affairs Division. This research is also supported in part by the Thailand Research Fund (TRF) and the Telecommunication Research and Industrial Development Institute (TRIDI). Thanks go to Associate Professor Manukid Parnichkun for providing the AIT test track for the experiment in this study. Special thanks go to friends and Createch Club members, in particular, Kalyarat Romaneeyangkul, Narisorn Limpaswadpaisarn, Somphop Limsoonthrakul, Tun Liangpaiboon, for their valuable comments and support in building the intelligent vehicles and data collection.

References

Ballard, D.H. and Brown, C.M. (1982). Computer Vision Prentice-Hall, Englewood Cliffs, NJ, USA.

Bribiesca, E. (1997). Measuring 2-D shape compactness using the contact perimeter. Computers and Mathematics with Applications, 33(11):1-9.

Carlbom, I. and Paciorek, J. (1978). Planar geometric projections and viewing transformations. ACM Computing Surveys (CSUR), 10(4):502.

Cui, Y.J. and Ge, S.S. (2003). Autonomous vehicle positioning with GPS in urban canyon environments. IEEE Transactions on Robotics and Automation, 19(1):15-25.

Deelertpaiboon, C. and Parnichkun, M. (2008). Fusion of GPS, compass, and camera for localization of an intelligent vehicle. International Journal of Advanced

Robotic Systems, 5(4):315-326.

Gonzalez, R.C. and Richard, E.W. (1992). Digital image processing. Addison Wisley.

Introduction - Google Earth user guide. (2009) Available from: http://earth.google.com/userguide/v4/.

Hayashi, M., Tanaka, T., and Yonekawa, M. (2008). High accuracy positioning of two-wheeled vehicle at high speed traveling using GPS. the SICE Annual Conference on Instrumentation, Control and Information Technology, Tokyo, Japan, p. 3500-3503.

Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J. (1993). Global positioning system. Theory and Practice.

Limsoonthrakul, S., Dailey, M.N., and Parnichkun, M. (2009). Intelligent vehicle localization using GPS, compass, and machine vision. the IEEE/RSJ International Conference on Intelligent Robots and Systems, Missouri, USA, p. 3981-3986

Miller, I., Campbell, M., Huttenlocher, D., Kline, F.R., Nathan, A., Lupashin, S., Catlin, J., Schimpf, B., Moran, P., and Zych, N. (2008). Team Cornell's Skynet: Robust perception and planning in an urban environment. Journal of Robotic Systems, 25(8):493-527.

Naranjo, J.E., Gonzalez, C., Garcia, R., Pedro, T.E., Revuelto, J., and Reviejo, J. (2004). Fuzzy logic based lateral control for GPS map tracking. the IEEE Intelligent Vehicle Symposium, Parma, Italy, p. 397-400.

Passino, K.M. (2004). Biomimicry for optimization, control, and automation. Springer Verlag.

Ratsamee, P., Surapa, T., and Kondo, T. (2010). An enhancement of waypoint navigation system for intelligent vehicle using Google Earth. the International Conference on Information and Communication Technology for Embedded Systems, Pathum Thani, Thailand.

Richard, B. (2005). Intelligent vehicle technology and trends. Artech House.

Siciliano, B. and Khatib, O. (2008). Springer handbook of robotics. Springer-Verlag New York Inc.

Sukkarieh, S., Nebot, E.M., and Durrant-Whyte, H.F. (1999). A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. IEEE Transactions on Robotics and Automation, 15(3):572-578.

Thailand Intelligent Vehicle Challenge. (2007). Available from: http://tivc.ait.ac.th.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M. and Hoffmann, G. (2006). Stanley: The robot that won the DARPA Grand Challenge. Journal of Field Robotics, 23(9):661-692.

Vincent, L. (1992). Morphological area openings and closings for grey-scale images. Shape in Picture: Mathematical Description of Shape in Grey-level Images, p. 197-208.

Webber, F. (2003). Statement of the alliance of automobile manufacturers. Presented before the Commerce, Science and Transportation Committee of the U.S. Senate, Noverber 15, 2005.

Xiangdong, L., Kirubarajan, T., Bar-Shalom, Y., and Xiaorong, L. (2001). Enhanced accuracy GPS navigation using the interacting multiple model estimator. the IEEE Aerospace Conference, Big Sky, MT , USA, p. 1911-1923.