DARPA GRAND CHALLENGE - 2004
SUMMURY OF SENSOR FUSION MODULE ON TERRAMAX

Zhiyu Xiang, Andy Chien 04/05/2004

Abstract

This report summarized the sensor fusion module for the TerraMax, including the sensors used, the performance they have, the fusion structures used to organize the sensors and the algorithms used for the fusion process. Some experimental results are presented. Some lessons learned and future works are pointed out as well.

I. SYSTEM OVERVIEW

1.1 TerraMax Hardware Configuration and Sensor Fusion System

Fig. 1 shows the hardware configuration of the TerraMax. In the view of sensors, we can divide it into three different groups: a). High level sensing group, which perceives the surrounding environment with some external sensors including mono-vision, stereo-vision, ladars, radars and sonars; b) Position and state sensing group, which provides the current GPS position and state of the truck; c) low level sensing group, which is composed of sensors measuring the actual controlling quantities, i.e., steering angle, throttle position, etc. Low level sensing will be introduced in the summary of control group, in this report we will focus on the other two sensing groups.

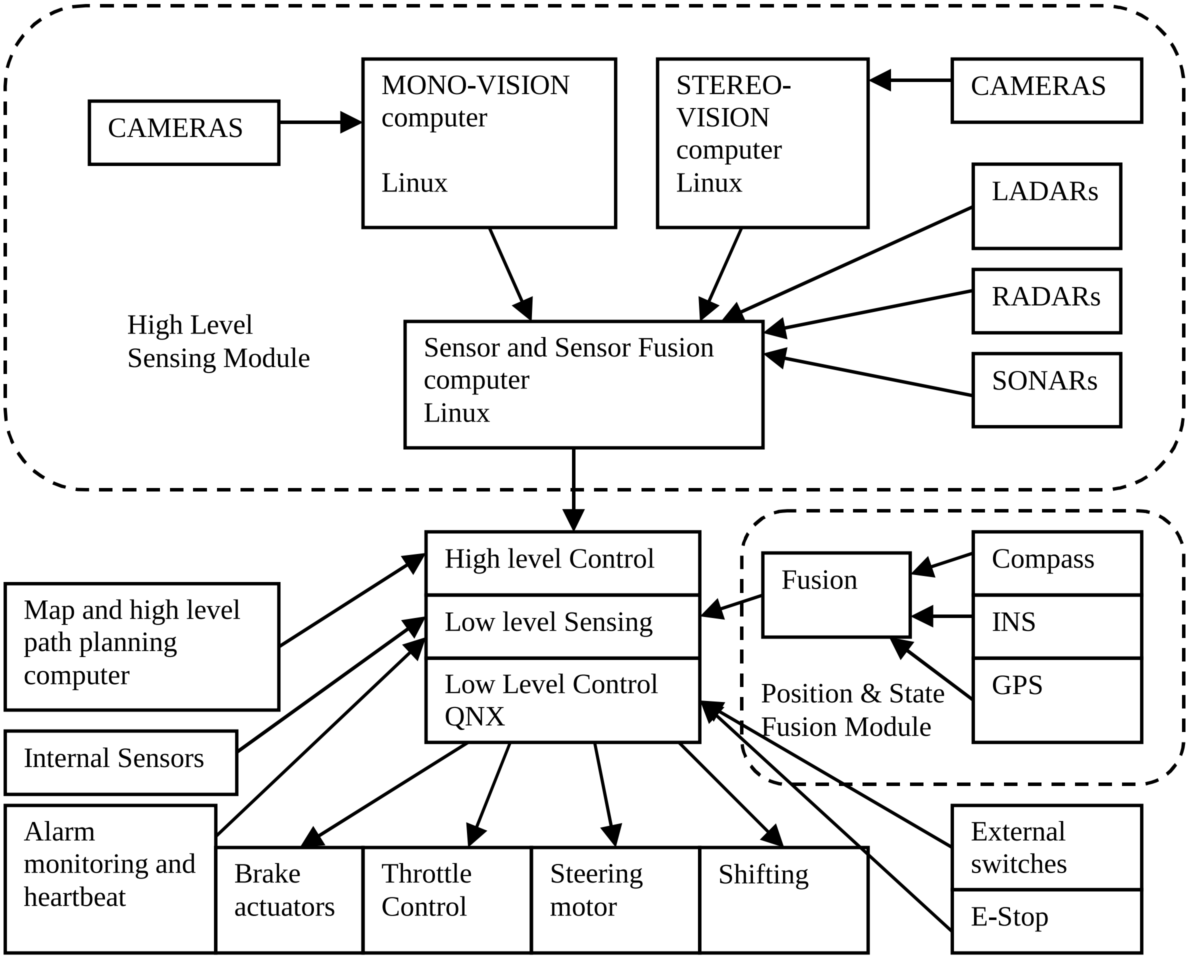


Fig.1 TerraMax Hardware Configuration

II. Position and State Fusion Module

2.1 Module Formation

Fig. 2 shows the GPS/INS/Compass fusion diagram. There are two GPS, one INS and one compass in the system. All of the datum from those sensors are collected through RS232 serial ports and fed into an Extend Kalman Filter based fusion algorithm to obtain the vehicle’s position and state.

2.2 Sensor performance

2.2.1 Novatel GPS system

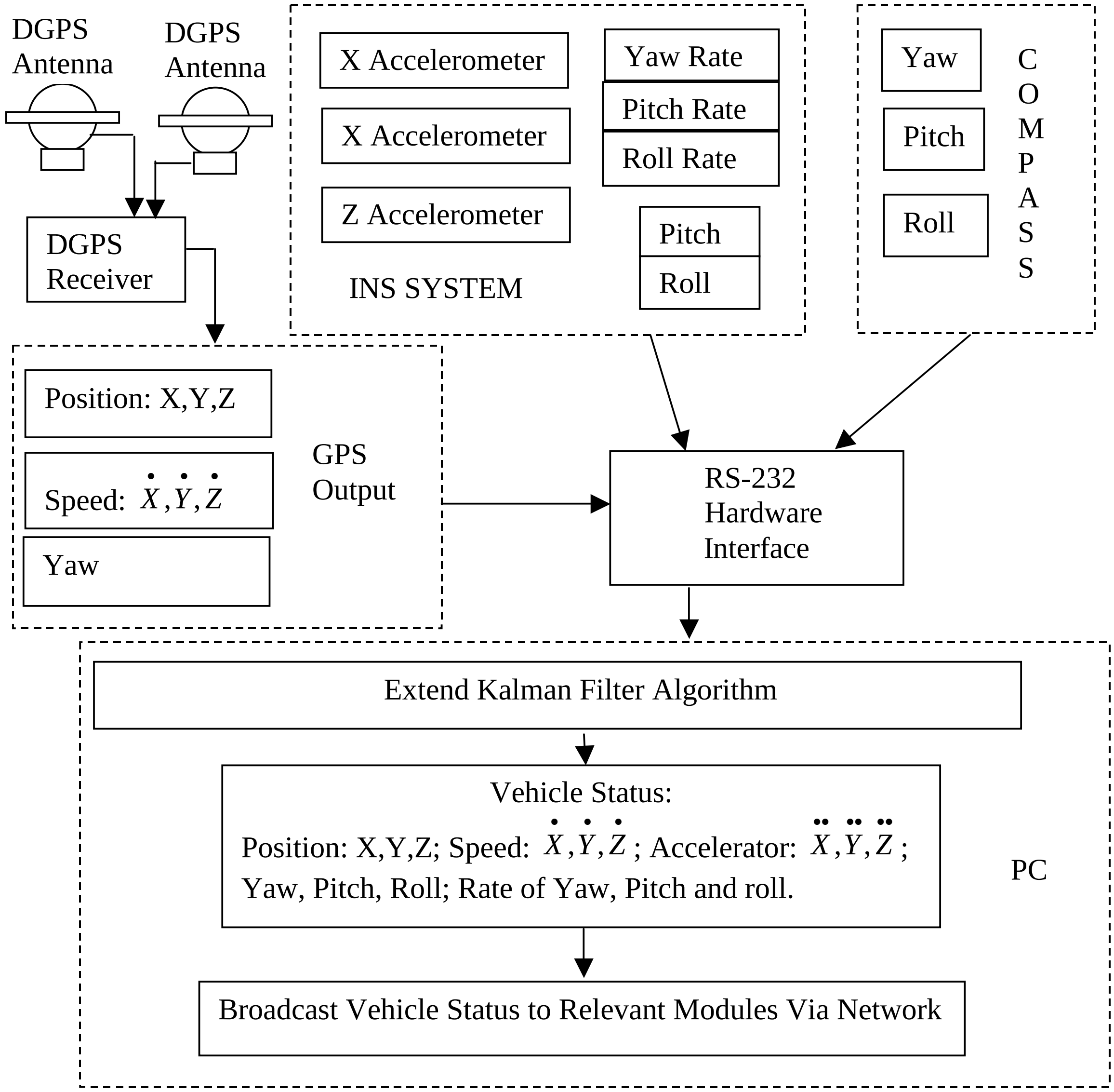

DGPS Antenna
DGPS Antenna
DGPS Receiver
X Accelerometer
X Accelerometer
Z Accelerometer
INS SYSTEM
Yaw Rate
Pitch Rate
Roll Rate
Pitch
Roll
Yaw
Pitch
Roll
COMPASS
Position: X,Y,Z
Speed: $\dot{X},\dot{Y},\dot{Z}$
Yaw
GPS Output
RS-232 Hardware Interface
Extend Kalman Filter Algorithm
Vehicle Status:
Position: X,Y,Z; Speed: $\dot{X},\dot{Y},\dot{Z}$; Accelerator: $\ddot{X},\ddot{Y},\ddot{Z}$;
Yaw, Pitch, Roll; Rate of Yaw, Pitch and roll.
Broadcast Vehicle Status to Relevant Modules Via Network
PC

Fig.2 The GPS/INS/Compass Fusion Diagram

With the OmniStar's High Precision (HP) service, the Novatel's ProPak-LB GPS receiver, which we used in our system, can reach very high position accuracy up to 10 cm with updating frequency up to 20 Hz. It can also provide velocity to an accuracy of 0.3m/s. Despite of these features which are important to UGV's navigation, some drawbacks of GPS are still need to be considered.

- The GPS can provide the direction of speed via calculating the difference of the position with last step. But when UGV is static or the speed is very low, the yaw angle obtained through this method is erroneous. Thus we need compass to provide the yaw angle anytime.

- GPS is a line-of-sight, radio navigation system, and therefore GPS measurements are subject to signal outages, interference, and jamming, whereas INS is a self-contained, non-jammable system that is completely independent of the surrounding environment, and hence virtually immune to external disturbance. Therefore, INS can continually provide navigation information when GPS experiences short-term outages.

2.2.2 Digital Compass

Electronic compass module HMR3000, can provide heading, pitch and roll output for navigation and guidance systems. This compass provides fast response time up to 20 Hertz and high heading accuracy of 0.5 deg with 0.1 deg's resolution. It provides the RS-232 interface for data communication. The output range of yaw angle is 0~360 deg with north and measurement value increases when rotating clockwise.

2.2.3 CrossBow INS (VG700AA-201)

VG700AA-201 can provide angular rates and accelerations plus pitch and roll. Data is available in Earth Coordinate, which makes the position fusing more easily. It has excellent bias stability of less than 20deg/hour and data updating frequency of more than 100 Hz.

2.3 Fusing Algorithm

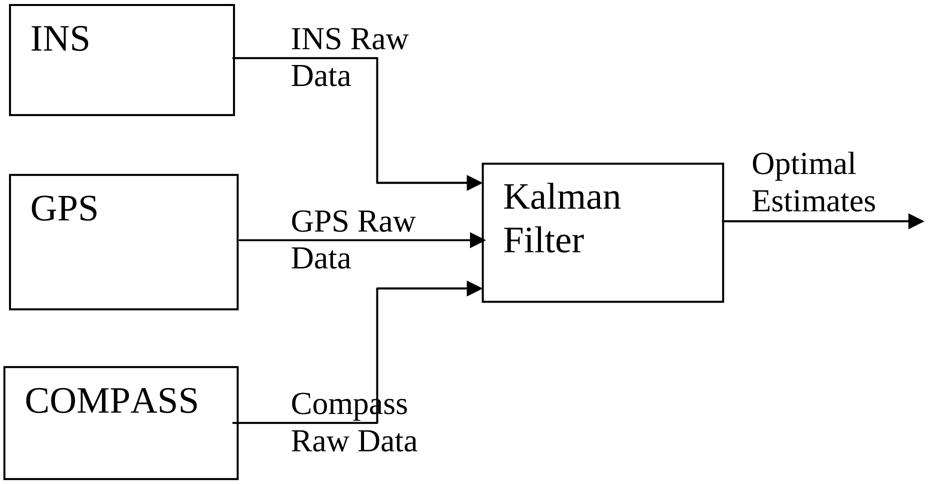


Fig. 3 Configuration of the State Estimator

As shown in Fig. 3, we adopt direct configuration of the kalman filter, which is, feeding the raw data to the Kalman filter and estimating all system state directly from the kalman filter.

Observation Equations of the Kalman filter:

$$\begin{cases} Z_e(k) = e(k) + w_e(k) \\ Z_n(k) = n(k) + w_n(k) \\ Z_h(k) = h(k) + w_h(k) \\ Z_{ve}(k) = V_e(k) + w_{ve}(k) \\ Z_{vn}(k) = V_n(k) + w_{vn}(k) \\ Z_{vh}(k) = V_h(k) + w_{vh}(k) \\ Z_\alpha(k) = \alpha(k) + w_\alpha(k) \\ Z_{\dot\alpha}(k) = \dot\alpha(k) + w_{\dot\alpha}(k) \\ Z_{ax}(k) = \cos\alpha(k) a_e(k) + \sin\alpha(k) a_n(k) + w_{ax}(k) \\ Z_{ay}(k) = -\sin\alpha(k) a_e(k) + \cos\alpha(k) a_n(k) + w_{ay}(k) \\ Z_{az}(k) = a_h(k) + w_{ah}(k) \end{cases} \qquad (1),$$

where capital letter Z means observations of the corresponding quantities, e, n, h and their subscripts represent the 3D coordinates in eastern, northern and elevation direction and V_e, V_n, V_h represent the corresponding velocity components respectively. a_e, a_n and a_h represent the acceleration components in earth surface coordinate system while a_x, a_y, a_z are the corresponding acceleration components in truck fixed coordinates respectively. α and $\dot\alpha$ represent the yaw angle and yaw rate.

State Equations for yaw and yaw rate:

$$X_\alpha(k+1) = \Phi_\alpha(k) X_\alpha(k) + \upsilon_\alpha(k) \qquad (2),$$

where:

$$X_\alpha(k) = \begin{bmatrix} \alpha(k) \\ \dot\alpha(k) \end{bmatrix},\ \Phi_\alpha(k) = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, R_\alpha(k) = \begin{bmatrix} \sigma_\alpha^2 & 0 \\ 0 & \sigma_{\dot\alpha}^2 \end{bmatrix}, Q_\alpha(k) = \begin{bmatrix} T_s^3\sigma_{\dot\alpha}^2/3 & T_s^2\sigma_{\dot\alpha}^2/2 \\ T_s^2\sigma_{\dot\alpha}^2/2 & T_s\sigma_{\dot\alpha}^2 \end{bmatrix}.$$

T_s is the time between two successive samples, R is the observation covariance matrix and Q is the processing noise covariance matrix.

State Equations for the state components along the eastern direction:

$$X_e(k+1) = \Phi_e(k) X_e(k) + \upsilon_e(k) \qquad (3),$$

where

$X_e(k) = \begin{bmatrix} e(k) \\ v_e(k) \\ a_e(k) \end{bmatrix}$, $\Phi_e(k) = \begin{bmatrix} 1 & T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$, $\upsilon_e(k)$ is the processing noise.

Similar to above equations, we can obtain the state equations for the rest states along the north and elevation direction.

Finally, the system state equation is like the following:

$$\begin{bmatrix} X_{\alpha}(k+1) \\ X_{e}(k+1) \\ X_{n}(k+1) \\ X_{h}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi_{\alpha} & 0 & 0 & 0 \\ 0 & \Phi_{e} & 0 & 0 \\ 0 & 0 & \Phi_{n} & 0 \\ 0 & 0 & 0 & \Phi_{h} \end{bmatrix} \begin{bmatrix} X_{\alpha}(k) \\ X_{e}(k) \\ X_{n}(k) \\ X_{h}(k) \end{bmatrix} + W(k) \qquad (4),$$

with the processing noise covariance matrix $Q(k) = \begin{bmatrix} Q_{\alpha} & 0 & 0 & 0 \\ 0 & Q_{e} & 0 & 0 \\ 0 & 0 & Q_{n} & 0 \\ 0 & 0 & 0 & Q_{h} \end{bmatrix}$ and

observation noise covariance matrix $R(k) = \begin{bmatrix} R_{\alpha} & 0 & 0 & 0 \\ 0 & R_{e} & 0 & 0 \\ 0 & 0 & R_{n} & 0 \\ 0 & 0 & 0 & R_{h} \end{bmatrix}$.

Equation (4) is the general state equation for the whole position fusion algorithm.

2.4 Experimental Results and Analysis

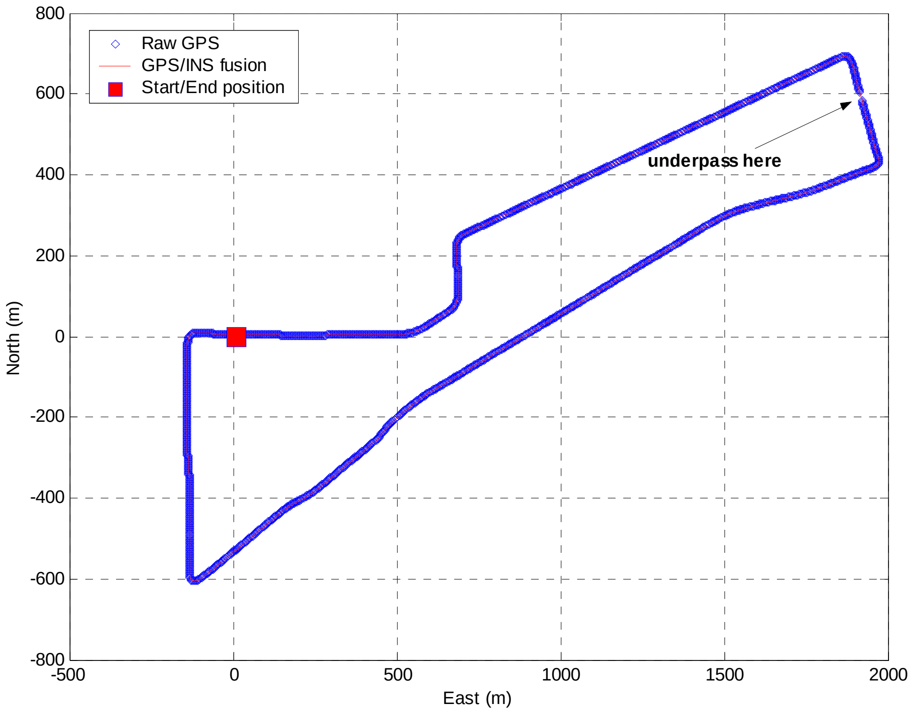


Fig. 4 Comparison of Raw GPS data and GPS/INS Fused Result

Fig.4 shows a test result of GPS/INS fusion. In this experiment, starting at around (9.2 m, -0.1 m), the truck was driven along a predefined loop, where an underpass was passed through. The position of the underpass was around (1914 m, 604 m). From Fig.4, the filter result is pretty good and the fusion algorithm can track the GPS data well when GPS signal was available and High Precision (HP) data was provided. During the underpass, the GPS outage happened, and INS output was used to estimate the truck's position. For clarification, this area was scaled larger in Fig.5.

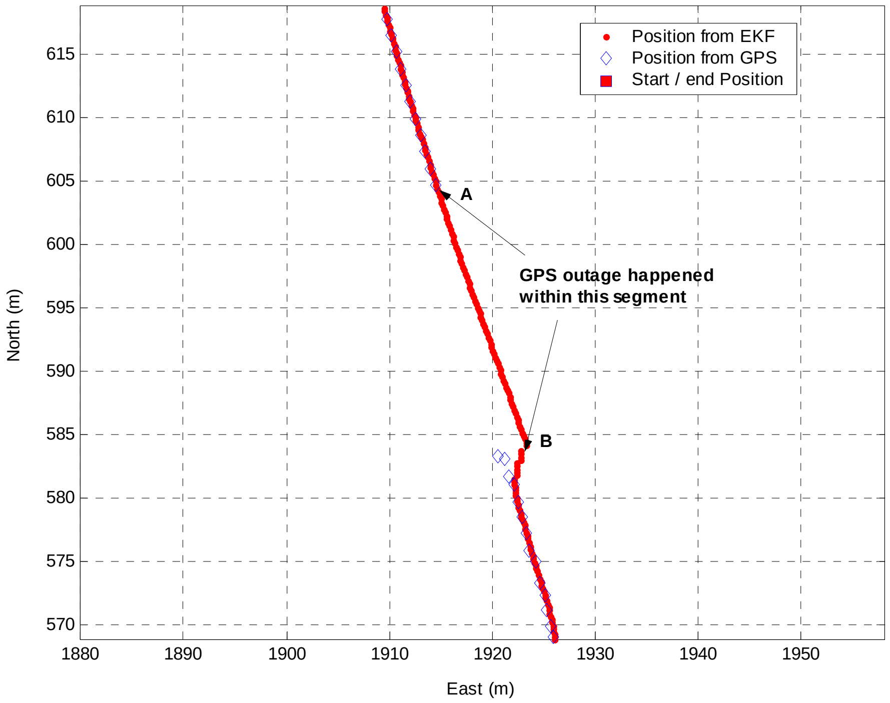


Fig. 5 The situation where GPS outage happens

From Fig. 5, under GPS outage, the fusion algorithm estimated the position of truck correctly. When GPS was available again, they matched each other well in the sense of small error boundary and the fused position adapted itself to the high precision GPS smoothly in short time.

Although the position results seems to be satisfied, the velocity results from the algorithm is not that good. As shown in Fig.7, the filtered velocity seems to be noisier than the original data from GPS. These noises are due to the measurement noise of acceleration from INS.

Part of the velocity's error is due to the yaw error, as shown in Fig.8. The heading angles from GPS and compass sometimes has 30 degree's difference, obviously in that case the compass data was erroneous. But why this happens is still under researching. Possible reason may be the insufficient calibration of compass or existence of something affecting the magnetic field of earth. Because we use compass's reading as the heading resources all the time, the filter's heading output is very close to the compass's input. Therefore in figure 8, the heading from the compass and the heading from the EKF are almost overlapped.

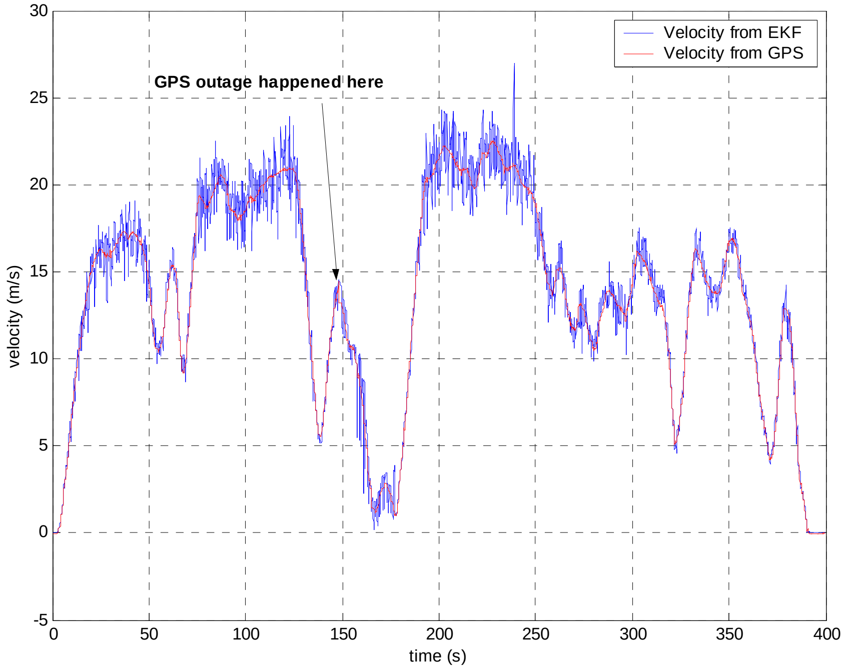


Fig. 7 The velocity from GPS and the fusion algorithm

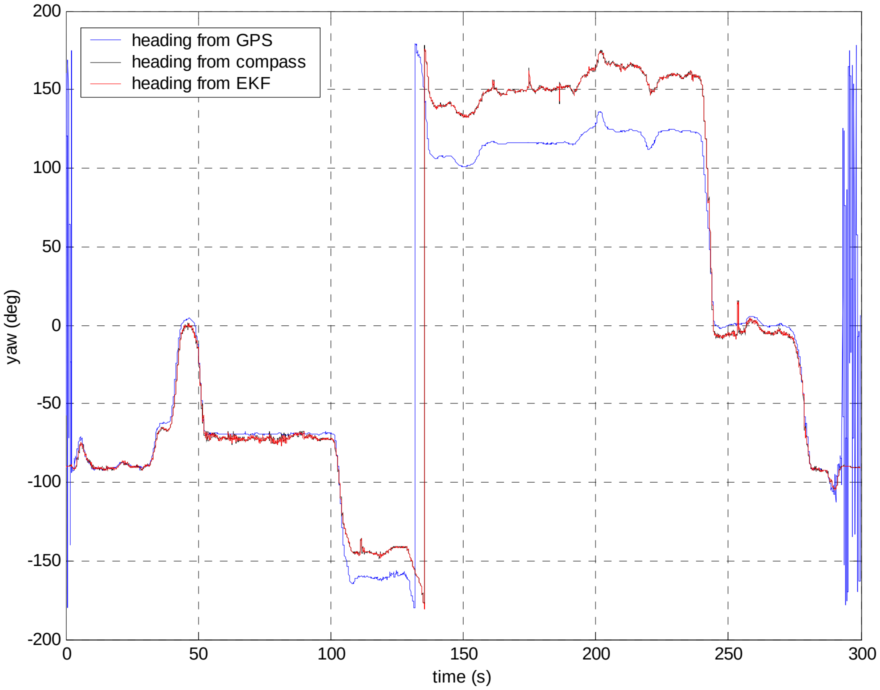


Fig. 8 Heading angles from GPS, compass and the EKF fusion output.

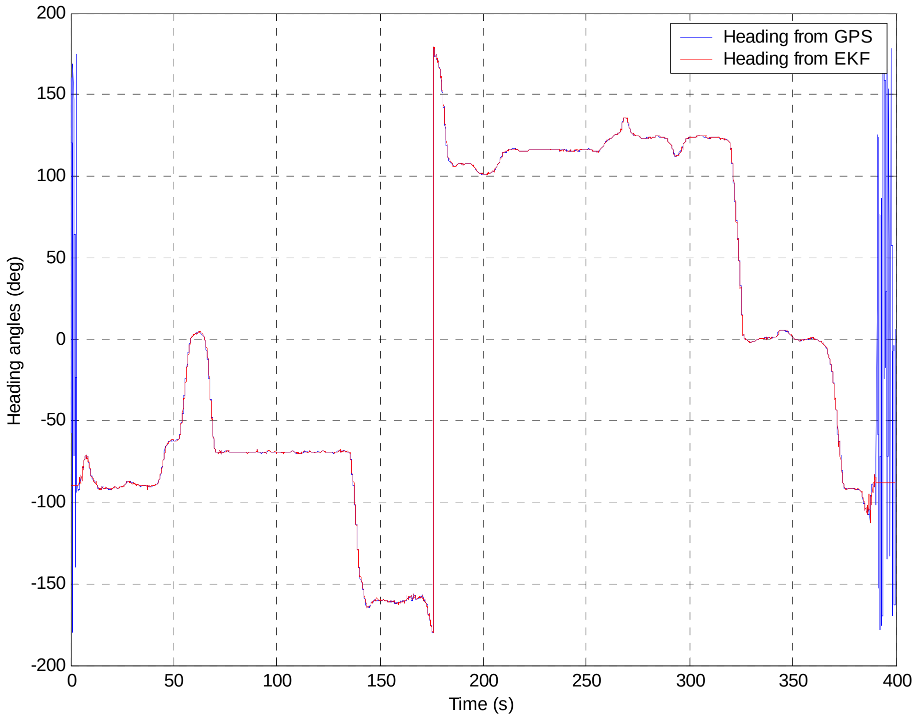


Fig. 9 Heading from GPS/ins yaw rate integration and from EKF output (almost overlapped together in most of the part).

However, if we only use the compass for initialization of the heading, and then use the GPS headings whenever GPS is available and use integrating value of yaw rate from INS during GPS intervals, the filter output becomes much better. As shown in Fig.9, they are overlapped together except the start and end of the experiment where the heading from GPS is erroneous due to the zero speed.

Similar to Fig. 7, in Fig.10 the acceleration from EKF is also noisier than that from the INS. One big noise at the time 150 is because the GPS recovery happened at that time and there is some position difference between the dead-reckoning position and the recovered GPS position, a self-tuning process is happened to track the GPS position again.

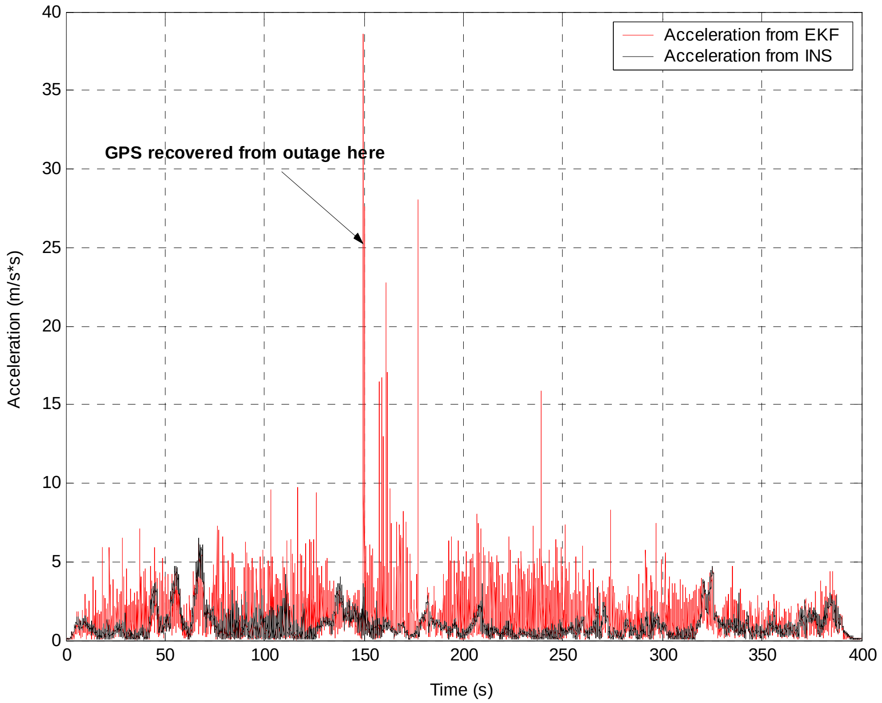


Fig. 10 The acceleration from GPS and the Extended Kalman Filter

2.5 Future works

Future works lie in two aspects:

(a) Do more tests on GPS/INS fusion under different conditions, including under more irregular terrain with higher speed and longer time of GPS outages, to see if the proposed GPS/INS fusion algorithm is robust enough and what factors affect the fusion result.

(b) The Kalman filter’s internal three order structure makes the tuning process difficult. Some parameters cannot be tuned independently to obtain the ideal filtering results for all system states.

Since the highly precise GPS (3 sigma within 15 cm) with high updating frequency (up to 20 Hz) is already available, actually it is no longer necessary to filter the GPS data in such way. The goal of algorithm becomes using good GPS readings directly as the position of the truck as long as GPS is available and only using estimated results otherwise. Therefore another way to improve the positioning system is detaching the position estimating from the original Kalman filter and just trying to obtain the optimal velocity from the GPS’s velocity in conjunction with INS’s acceleration input. When GPS outage happens, just use the optimal velocity to dead-reckon the position of the truck. It is easier to implement and more practicable.

III. HIGH LEVEL SENSOR FUSION MODULE

3.1 Introduction of Sensor Fusion System

3.1.1 Ladar

Ladar is a range measurement system based on advanced laser techniques. With its 180 degree's scanning scale, it provides the range between the origin of sensor and obstacles in a resolution up to 1 cm. Four Ladars are induced in our system to detect obstacles ahead of the UGV. Three are mounted to scan horizontally, detecting the positive obstacles around the UGV and one is mounted vertically, measuring the negative obstacles and slopes.

3.1.2 Radar

Radar is able to detect obstacles in longer distance and works well under bad weather conditions, i.e., snow, fog or rain. With its build-in target tracking technique, it can track the moving obstacles ahead stably, providing their position and velocity simultaneously. Comparing with ladar, these are two exciting merits of radar. On the other hand, radar has narrow beam width, which leads to unable to detect the near vehicles on the neighboring lane.

3.1.3 Sonar

Sonar is used to detect short range obstacles. More sonars can be installed around the UGV due to its good accuracy and low cost. Sonar can also help detecting one side fall terrains.

3.1.4 Vision

Vision is composed of two independent parts: mono-vision and stereo-vision. Mono-vision is responsible for road finding while stereo vision is responsible for positive obstacle detecting. They are designed to find free space or trails under off-road condition.

3.2 The Fusion Hierarchy

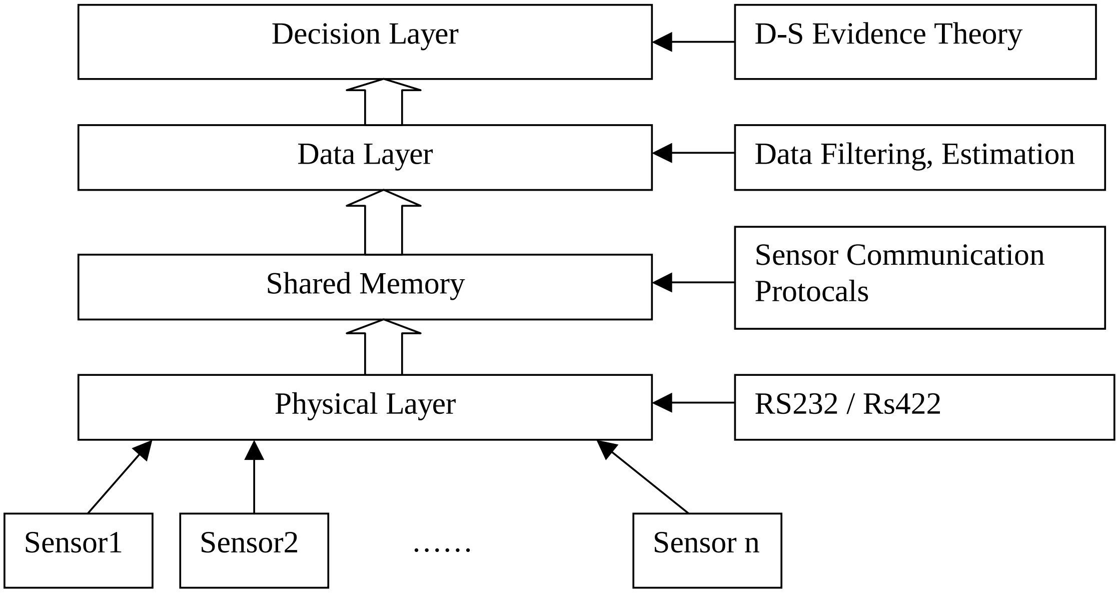


Fig. 11 The hierarchy structure of sensor fusion system

As fig.11 shows, the proposed sensor fusion system can be divided into four different layers. From physical layer to decision layer, the information is becoming more and more "concentrated" in the sense of perception and understanding. Different processing algorithm is used in each layer.

3.3 Information flow of the sensor fusion system

The sensors' mounting scenario on Terramax can be found in Appendix A. Here the information flow of the sensor fusion system is illustrated in Fig 12.

3.4 Fusion Map Formation

As shown in Fig. 13, a grid-based map is set up for sensor fusion and environmental description. The map is 100 m X 100 m's large with 50 meters wide on each side of the truck. Based on the information from sensors, each grid can be arranged with "occupied", "empty" or "unknown" together with the confidence value. The map is north oriented, taking the center of the truck as the origin. Therefore the origin of the map is moving with the truck, while the orientation is constant. With time goes on, the information from the sensors can be "overlapped" onto the map, which finally leads to a more accurate, more trustable description of the environment.

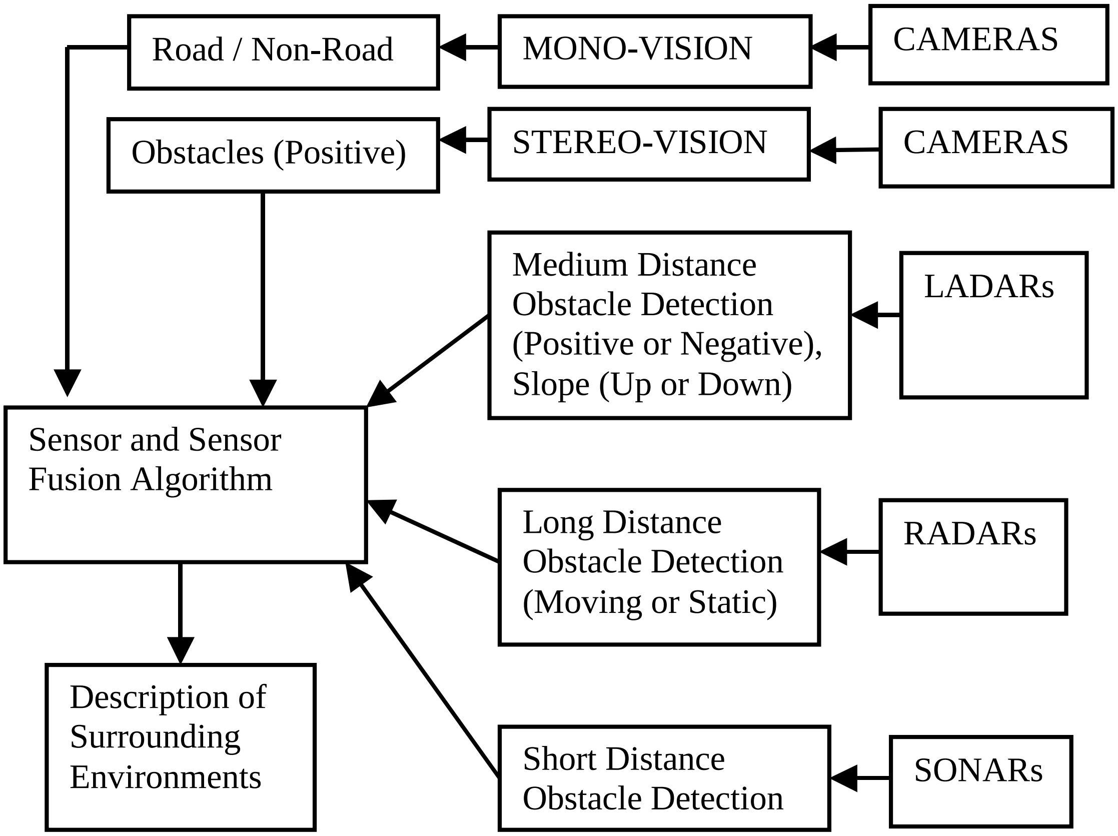

Road / Non-Road
MONO-VISION
CAMERAS
Obstacles (Positive)
STEREO-VISION
CAMERAS
Medium Distance Obstacle Detection (Positive or Negative), Slope (Up or Down)
LADARs
Sensor and Sensor Fusion Algorithm
Long Distance Obstacle Detection (Moving or Static)
RADARs
Description of Surrounding Environments
Short Distance Obstacle Detection
SONARs

Fig. 12 The information flow of the fusion algorithm

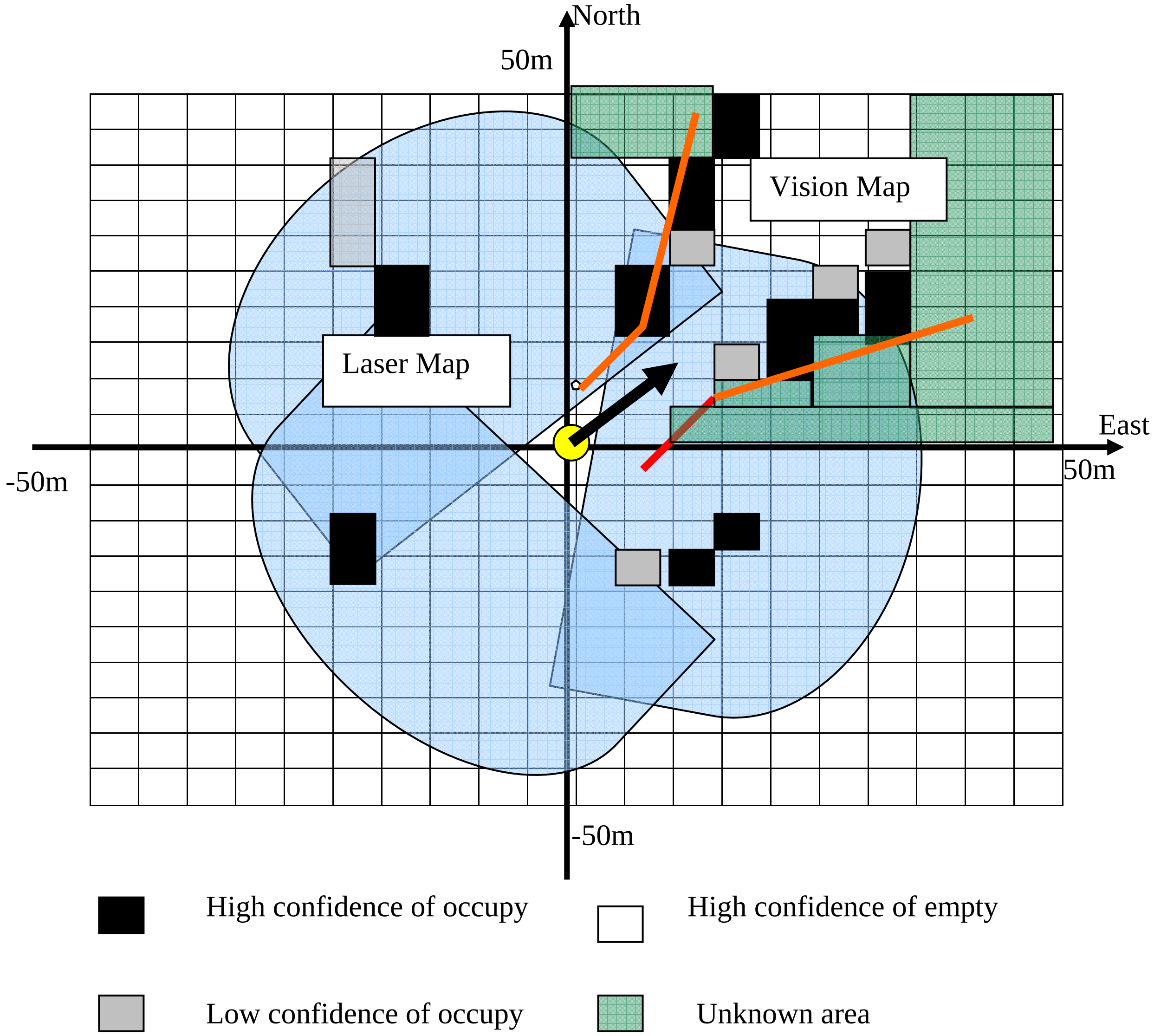


Fig. 13 The formation of the map

Some parameters of the proposed map is as the follows:

a. Input from vision (125x125 cells, 1 byte/cell)

Bits	Cell Data
0~2	Objects Data
3~6	Confidence
7	Reserved

000 Road
001 Cover
010 Positive Obstacles
011 Negative Obstacles
110 Moving Obstacles
111 Unknown Area

b. Output format (250 x 250 cells, 1byte/cell)

Bits	Cell Data
0~2	Objects Data
3~4	Confidence
5~7	Indices of Moving Obstacles

000	Road
001	Cover
010	Positive Obstacles
011	Negative Obstacles
110	Moving Obstacles
111	Unknown Area

c. Resolution: 0.4 m X 0.4m;
d. Orientation: North;
e. Origin: Truck center.
f. Updating frequency: 10 Hz.

3.5 The Fusion Algorithm

3.5.1 Map updating
Shown in Fig. 14.

3.5.2 Deducing Algorithm

As mentioned before, we try to integrate the information from different sensors into a current sensor map and accumulate the sensor map formatted at different time into the fusion map to obtain a better understanding of the surrounding environment. As expected, conflictions can happen among sensor readings and between the sensor map and the fusion map. How to deal with it? How to obtain a reasonable and good understanding from the conflicted sensor readings? We adopt a Dempster-Shafer Evidence deducing machine to deal with it. Please find the deducing table in **Appendix B**.

3.6 The Minimum Sensor Fusion System and the Grand Challenge

Due to some reasons, we designed a minimum sensor fusion system for the TerraMax, which is composed of two forward-looking LADARs only. Generally, it works well during the Grand Challenge in the sense of obstacle detection. However, it is only a very simplified sensor fusion system with very limited information yielded.

Our truck was stuck within the bushes after 1.2 miles away from the starting line, the possible reason lies in the following aspects:

1. The free space between the bushes is incorrect and seems to be too narrow to pass through. Possible reasons:

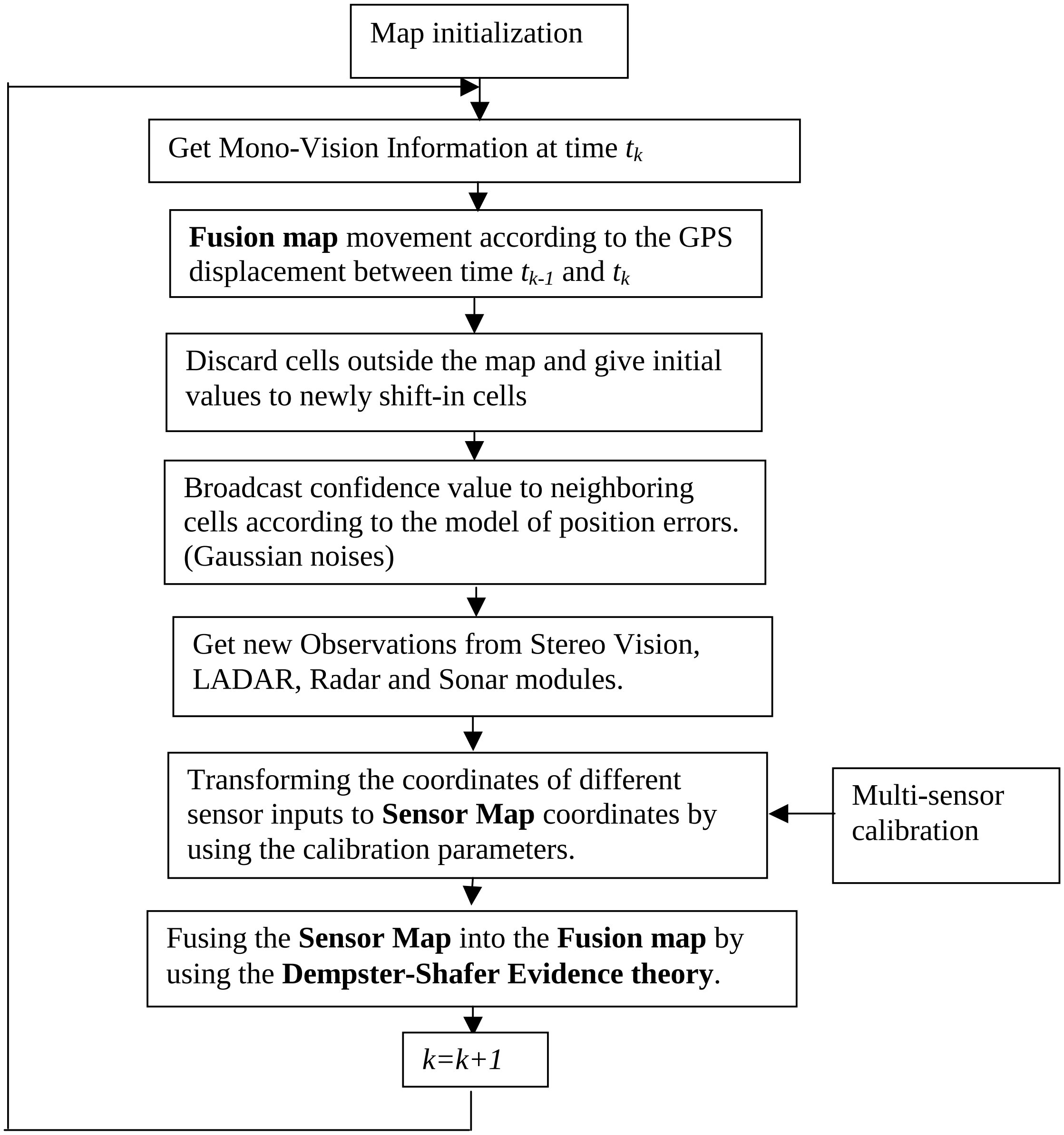

Map initialization
Get Mono-Vision Information at time t_k
Fusion map movement according to the GPS displacement between time t_{k-1} and t_k
Discard cells outside the map and give initial values to newly shift-in cells
Broadcast confidence value to neighboring cells according to the model of position errors. (Gaussian noises)
Get new Observations from Stereo Vision, LADAR, Radar and Sonar modules.
Transforming the coordinates of different sensor inputs to **Sensor Map** coordinates by using the calibration parameters.
Multi-sensor calibration
Fusing the **Sensor Map** into the **Fusion map** by using the **Dempster-Shafer Evidence theory**.
$k=k+1$

Fig. 14 Map Updating Algorithm

a. The resolution of 0.4 meter for the fusion map is not high enough, upgrading to 0.2 meter may help. But decreasing the grid size to a half means increasing the computational loads 4 times. This may degrade the output rate to 2.5 Hz if the same computer is used. Therefore a computer with better configuration is needed.

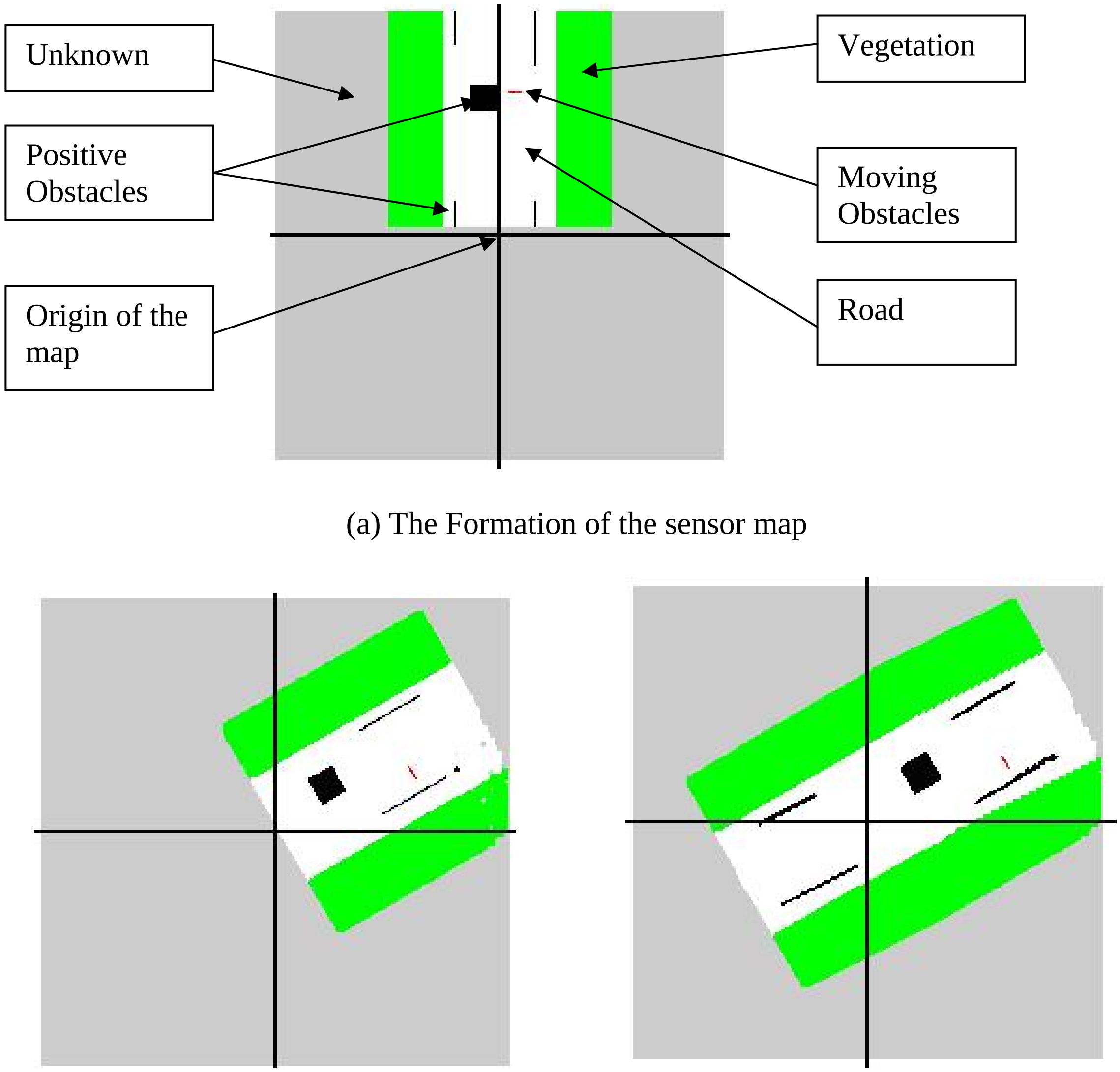
b. The high level control module expanded the obstacles too much in the fusion map for safety reasons.

2. The Ladars scan horizontally and are able to detect any obstacles higher than the scanning plane. Although the scanning plane is carefully tuned, a tradeoff has to be made between the safety of the truck and the unnecessary bush clusters. If the

stereo-vision is available, theoretically it knows the height of obstacles, based on which the obstacles lower than a threshold can be deleted.

3.7 Experimental Results

Fig. 15 shows some simulation results from the sensor fusion module. Fig. (a) illustrates the formation of sensor map. This map is set up in the truck frame, taking the current heading direction as the y-axis. The perceived environmental information on current moment, including road, vegetation coverage, positive obstacles, negative obstacles and moving obstacles, is integrated into it. After rotating the sensor map with the heading angle, the sensor map is integrated into the existed fusion map by utilizing the Deducing table shown in Appendix B. The fusion map is shown in (c).



(a) The Formation of the sensor map

(b) Rotated sensor map according to the heading angles

(c) The fusion results (fusion map).

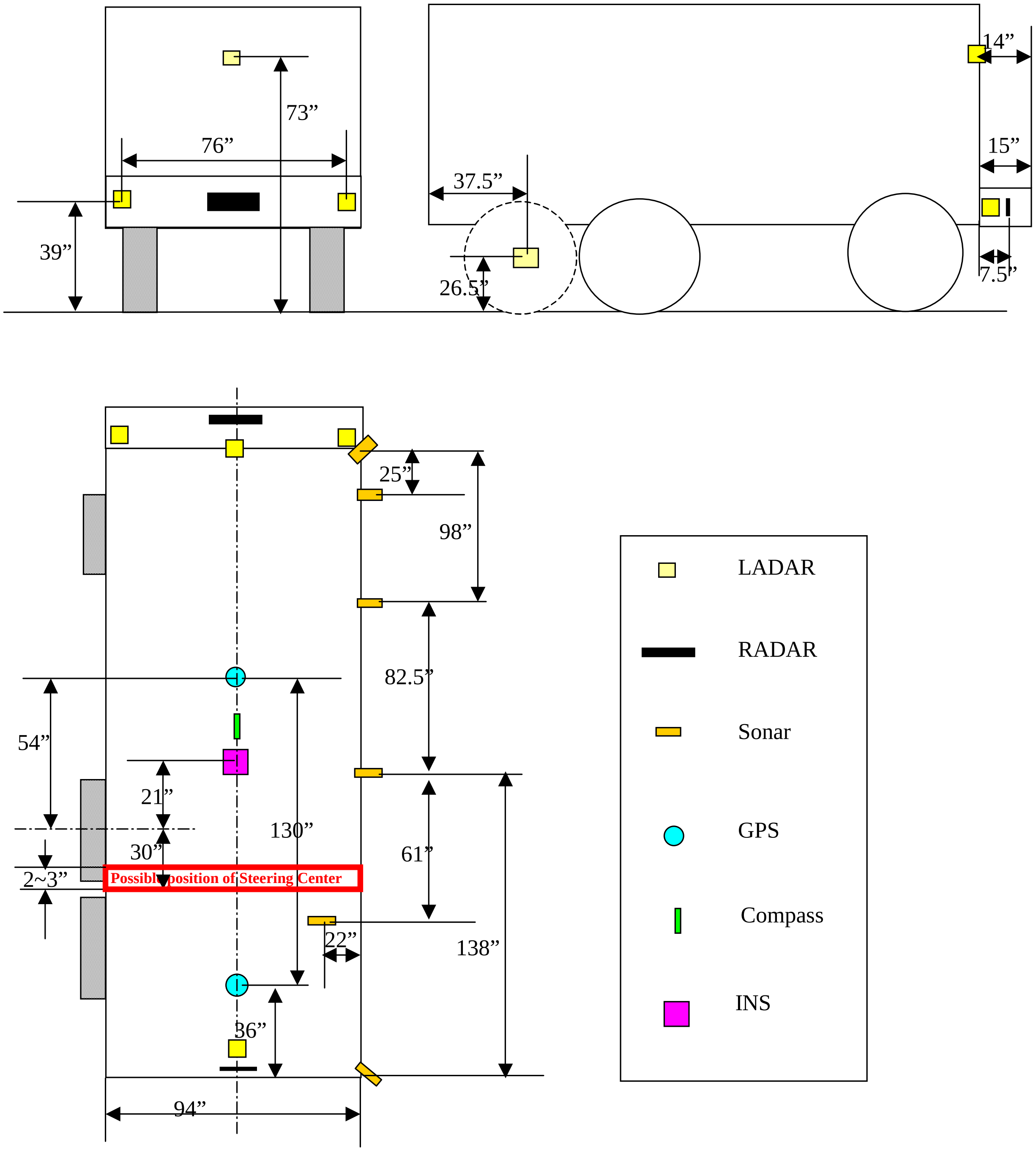
Fig. 15 The simulation results of the sensor fusion

3.8 Future Works

Future works lie in the following aspects:

1. Increasing the resolution of map to 0.2 meter, resizing the map down to 60 square meters with 30 meter's wide in each side. A computer with better computing performance is needed;
2. Setting up a special small map for robotic motion, covering 3 meters' space around the truck by using ultrasonic and laser sensors;
3. Integrating all of the sensors into the system to obtain more results;
4. Finding better ways to tell the bushes from the common obstacles, possibly infrared camera, radar or other sensors can be used.

Appendix A: The Mounting of the Sensors on TerraMax



Appendix B: The Deducing Table for Sensor Fusion Algorithm

		Information from **S**ensor Map					
Information from **F**usion Map		ROD m_e_s	COV m_e_s	POB m_o_s	NOB m_o_s	MOB m_o_s	UKN m_u_s
	ROD m_e_f	ROD (1)	COV if (a) ROD if (b) (1)	POB if (g), results (3); ROD if (h), results (5).	NOB if (g), results (3); ROD if (h), results (5).	MOB (3)	ROD (2)
	COV m_e_f	ROD if (a) COV if (b) (1)	COV (1)	POB if (g), results (3); ROD if (h), results (5).	NOB if (g), results (3); COV if (h), results (5).	MOB (3)	COV (2)
	POB m_o_f	ROD if (e), results (3); POB if (f), Results (4).	COV if (e), results (3); POB if (f), results (4).	POB (1)	NOB if (c) POB if (d) (1)	MOB (3)	POB (2)
	NOB m_o_f	ROD if (e), results (3); NOB if (f), results (4)	COV if (e), Results (3); NOB if (f), Results (4).	POB if (c) NOB if (d) (1)	NOB (1)	MOB (3)	NOB (2)
	MOB m_o_f	No prediction exists in the fusion map, replaced by UKN.					
	UKN m_u_f	ROD (3)	COV (3)	POB (3)	NOB (3)	MOB (3)	UKN (3)

ROD---Road; COV---Cover; POB---Positive Obstacle; NOB---Negative Obstacle; MOB---Moving Obstacle; UKN---Unknown area.

m -- confidence, o --occupied, e --empty, u -- unknown, subscript f --fusion map, subscript s --sensor map. For example, m_o_f represents the confidence of occupying for a special grid on fusion map.

Conditions:

(a): $m\_e_s > m\_e_f$; (b): $m\_e_s <= m\_e_f$.

(c): $m\_o_s > m\_o_f$; (d): $m\_o_s <= m\_o_f$.

(e): $m\_e_s >= m\_o_f + m\_u_f$; (f): $m\_e_s < m\_o_f + m\_u_f$.

(g): $m\_o_s >= m\_e_f + m\_u_f$ or a cell signed “OB(POB/NOB)” exists in 3x3 neighboring area;

(h): $m\_o_s < m\_e_f + m\_u_f$ and no cell signed “OB” exists in 3x3 neighboring area.

Results:

(1): D-S Evidence Deducing: $\begin{cases} m\_e'_f = k \sum\limits_{X_1 \cap X_2 = empty} (m_1(X_1) * m_2(X_2)) \\ m\_o'_f = k \sum\limits_{X_1 \cap X_2 = occupy} (m_1(X_1) * m_2(X_2)) \\ m\_u'_f = k \sum\limits_{X_1 \cap X_2 = unknown} (m_1(X_1) * m_2(X_2)) \end{cases}$, where $k = (1 - \sum\limits_{X_1 \cap X_2 = \Phi} (m_1(X_1) * m_2(X_2)))^{-1}$

(2): Keep information in the fusion map:

$$m\_e'_f = m\_e_f,\ m\_o'_f = m\_o_f,\ m\_u'_f = m\_u_f.$$

(3): Update the fusion map with the information from the sensor map:

$$m\_e'_f = m\_e_s,\ m\_o'_f = m\_o_s,\ m\_u'_f = m\_u_s.$$

(4): Original certainty decreased, keeping the OB classification:

$$m\_o'_f = m\_o_f - 10,\ m\_e'_f = m\_u_f = (1 - m\_o'_f)/2.$$

(5): Original certainty decreased, keeping the ROD or COV classification:

$$m\_e'_f = m\_e_f - 10,\ m\_o'_f = m\_u_f = (1 - m\_e'_f)/2.$$