MATEC Web of Conferences 30, 04003 (2015)
DOI: 10.1051/matecconf/20153004003
© Owned by the authors, published by EDP Sciences, 2015

Sensor Fault Detection and Diagnosis for autonomous vehicles

Miguel Realpe[1,2,a], Boris Vintimilla[2], Ljubo Vlacic[1]

[1]*Intelligent Control Systems Laboratory, Griffith University.Brisbane, Australia*
[2]*CIDIS - FIEC, Escuela Superior Politecnica del Litoral.Guayaquil, Ecuador*

Abstract. In recent years testing autonomous vehicles on public roads has become a reality. However, before having autonomous vehicles completely accepted on the roads, they have to demonstrate safe operation and reliable interaction with other traffic participants. Furthermore, in real situations and long term operation, there is always the possibility that diverse components may fail. This paper deals with possible sensor faults by defining a federated sensor data fusion architecture. The proposed architecture is designed to detect obstacles in an autonomous vehicle's environment while detecting a faulty sensor using SVM models for fault detection and diagnosis. Experimental results using sensor information from the KITTI dataset confirm the feasibility of the proposed architecture to detect soft and hard faults from a particular sensor.

1 Introduction

Many autonomous vehicles are currently being tested on public roads in order to demonstrate safe and reliable operation in real world situations. Furthermore, fault-tolerant architectures have been reported for steering, braking, control and some specific sensor functions that integrate autonomous vehicles. However, long term behaviour of diverse sensors has not been tested and fault-tolerant perception architectures have not yet been developed.

The concept of fault tolerant systems refers to the systems that are able to compensate faults in order to avoid unplanned behaviours[1]. With the purpose of achieving this goal, a fault tolerant system should have the capability to detect and isolate the presence and location of faults, and then reconfigure the system architecture to compensate for those faults (fault recovery). Several sensor validation methods have been proposed on diverse applications. Some sensor validation methods produce their own health information using a sensor alone. Usually, the sensor readings are compared to a pre-established nominal value and a faulty sensor is declared whenever a threshold value is exceeded.

A more common sensor validation method for complex systems is the analytical validation, which is based on information from multiple sensors. An analytical validation requires a model of the system or of the relation between the sensors, which is executed in parallel to the process and provides a group of features. Then, these features are compared with the system forming residual values. The residuals that differ from the nominal values are called symptoms and can be subject to a symptom-fault classification in order to detect a fault and its location (fault diagnosis) [1]. Model based methods are categorized as parity equations [2, 3], parameter estimation methods [4], and observer-based methods with Luenberger observers [5] or Kalman filters [6]. These methods are very popular for fault tolerant control systems. Nevertheless, soft computing techniques, such as neural networks, fuzzy logic, evolutionary algorithms and support vector machines (SVM), have been developed for fault detection and fault isolation, because it is not always possible to obtain a good model of the systems [7].

Fault diagnosis is based on observed symptoms and experience-based knowledge of the system [1]. One approach is the use of classification methods, where the relation between symptoms and faults are determined experimentally in a previous phase of the system. Another approach is the use of inference methods, where causal relations are created in the form of rules based on partially known relationships between faults and symptoms.

After identifying faults, a reconfiguration of the system architecture is required. Fault recovery can be achieved using direct redundancy or analytical redundancy [8]. With direct redundancy, a spare module is employed to replace the faulty one. Despite the fact that direct redundancy is effective and easy to configure, it can be very expensive and unfeasible. On the other hand, analytical redundancy implies utilizing the working modules to complete the tasks which failed. For instance, if there is a fault in a laser scanner of an autonomous vehicle, the information from two cameras can be used instead to create range data and compensate for the laser scanner functions.

In recent years, only a few specific solutions of fault tolerant perception systems for autonomous vehicles have been developed. However, many researchers have

[a]Corresponding author: mrealpe@fiec.espol.edu.ec

Article available at http://www.matec-conferences.org or http://dx.doi.org/10.1051/matecconf/20153004003

implemented fault tolerant modules in autonomous vehicles in areas such as vehicle navigation sensors. Furthermore, the multi-sensor architecture of navigation systems can be compared with perception systems. In general, two different architectures are applied for navigation systems; centralized architecture, which is a one-level fusion process with little fault tolerance against soft sensor faults [9] and federated architecture, which is a two-level fusion method with good fault tolerance potential.

Federated architecture is proposed by Carlson [10] to fuse decentralized navigation systems with the objective of isolating faulty sensors before their data becomes integrated into the entire system. This architecture is composed of a group of local filters, that operate in parallel, and a master filter (Figure 1). A fundamental component of the federated filter is the reference sensor. Its data is frequently used to initialise sensors and set pre-processing information in local filters. Consequently, the most reliable and accurate sensor should be chosen as the reference for the local filters [11]. In [12, 13] a federated Kalman filter is implemented in a multi-sensor navigation system, and a fuzzy logic adaptive technique is applied to adjust the feedback signals on the local filters and their participation in the master filter. Similarly, an expert system is implemented in [14] to adjust the information sharing coefficients for local filters.

In the present paper, a federated sensor data fusion architecture is proposed in order to provide fault tolerance to one of three redundant sensors of an autonomous vehicle's perception system. The architecture is then tested using single sensor hard and soft faults. This paper is organized as follows: Section II describes the proposed model, experimental results are shown in section III and conclusions are presented in section IV.

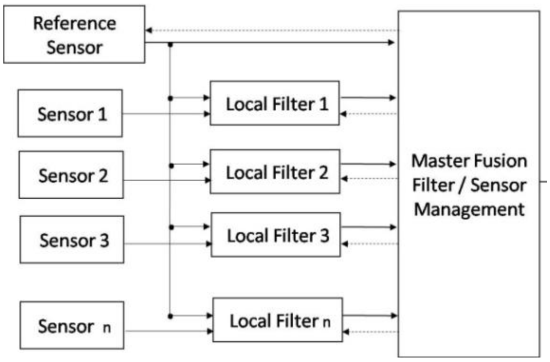


Figure 1. Federated sensor data fusion architecture.

2 Model Description

The proposed perception system is based on the Joint Directors of Laboratories model - JDL [15, 16], which is the most widely used model by the data fusion community. The JDL model is integrated by a common bus that interconnects five levels of data processing. A revision of the JDL model, the ProFusion2 - PF2 functional model, is proposed to apply sensor fusion in multi sensor automotive safety systems in [17]. It groups the original levels into three layers to add hierarchical structure. Also, it establishes inter-level and within-layer interactions, excluding the process refinement (level 4) from the original JDL model, which is related to resource management and monitors the overall data fusion process and provides a feedback mechanism to each of the other layers.

The present research proposes the re-integration of the process refinement level to the sensor fusion, which communicates with all levels, while maintaining the hierarchical structure of the PF2 functional model, as shown in Figure 2. In this model, the perception layer provides state estimations of the objects; the decision application layer predictsfuture situations and deduces output of potential manoeuvres; and the action/HMI layer collects and provides information to the user. Meanwhile, the process refinement layer analyses residuals from all the layers and provides information about faulty states to the decision application layer and feedback to each layer in order to minimize the effects of faults.

The implementation of the perception system has been done based on the perception sensors available in the KITTI dataset [19-21], which includes a Velodyne sensor and two pairs of stereo vision cameras. The federated perception architecture suggested to fuse sensor data from the KITTI dataset is shown in Figure 3. The system has been divided into different modules: one object detection for each sensor type, one local fusion for each support sensor, one master fusion, a tracking module and the Fault Detection and Diagnosis [FDD] module.

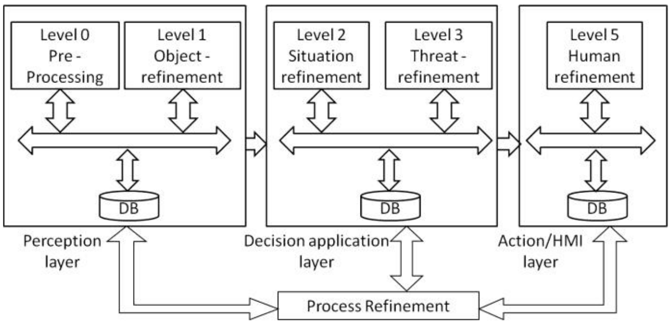

Level 0 Pre - Processing
Level 1 Object - refinement
Level 2 Situation refinement
Level 3 Threat - refinement
Level 5 Human refinement
DB
DB
DB
Perception layer
Decision application layer
Action/HMI layer
Process Refinement

Figure 2.Data fusion model for fault tolerant implementation[18].

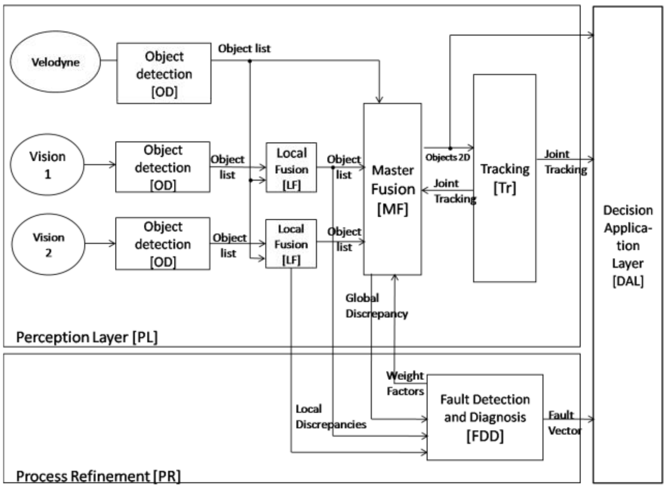


Figure 3. Fault Tolerant perception system for KITTI dataset[18].

2.1 Object Detection and Local Fusion

Object detection [OD] and local fusion [LF] have been implemented and described in [18]. Vision OD processes information from the cameras, combining motion detection, histogram of oriented gradients (HOG) detector and disparity maps in order to detect obstacles in the frontal area of the vehicle. On the other hand, Velodyne OD groups the scanned points into objects, according to their distances using the nearest neighbour algorithm based on the Euclidean distance metric. LF module creates an objects single list using data from a specific sensor and the reference sensor; it also creates the discrepancy values between those sensors, which represent the residuals used by the FDD module to determine the presence of a sensor fault.

2.2 Master Fusion

Master Fusion [MF] combines data from the reference sensor, LF modules and the tracking module. First, the lists of objects from all the inputs are fused based on the amount of their overlapping areas, creating candidate objects. Then, patterns in the objects' pixels and the weight from each sensor are used to validate pixels in the candidate objects. The discrepancies values from MF are estimated obtaining the difference between the numbers of pixels from the candidate objects list, the reference sensor and the fused objects list.

The pixel relationship of the objects is represented by a vector composed of six features. The first three features are boolean values that represent the origin of the object (reference, LF1, LF2) while the next three features are the distance fields values that show the distance of the corresponding pixel to the closest object. Also, three extra features representing the weight of each sensor are added in order to create a training vector (Table 1). The Master Fusion feature vector is trained offline with a SVM algorithm [22, 23] using positive vectors from a group of pixels that have been manually marked as detected objects and using negative vectors selected randomly from the other pixels (no objects).

Table 1. Master Fusion feature vector.

Feature	Value
Reference Sensor	True,False
Local Fusion 1	True,False
Local Fusion 2	True,False
Reference distance field	0-255
Local distance field 1	0-255
Local distance field 2	0-255
Weight reference	high, low, off
Weight vision 1	high, low, off
Weight vision 2	high, low, off

2.3 Fault Detection and Diagnosis

Fault Detection and Diagnosis [FDD] module applies SVM to recognize the changes in the discrepancies values from MF and LF modules. The LF discrepancy values are integer numbers representing the percentage of pixels from a fusion module that are present in its associated sensor and the reference sensor. For example, figure 4 shows the discrepancy from a local fusion module coded by colours: green represents pixels from the Velodyne OD, red represents pixels from the vision OD and yellow represents pixels that are present in both. On theother hand, the MF discrepancy is given by the difference between the resulting fused objects and objects detected by the reference sensor.

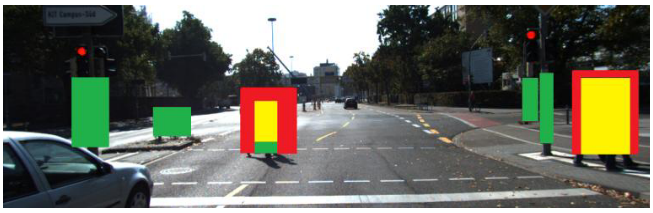


Figure 4. Discrepancy map from local fusion [18].

A SVM model is created for every sensor. Each model is trained using a vector of 9 features as shown in Table 2.The negative vectors are created introducing a displacement in the calibration matrix of the associated sensor, while the positive vectors are obtained from the unaltered data.

Table 2. FDD feature vector

LF1	Reference
	Vision 1
	Both
LF2	Reference
	Vision 1
	Both
MF	Reference
	Not Reference
	Fused

The FDD module has been trained to detect faults in a specific sensor. Thus, the system has 3 different models, one from each sensor and the faulty sensor is obtained directly from a specific model.

In order to avoid false positives the output from the SVM is consider only if a faulty response is given after N consecutive outputs. Then, the respective sensor is reconfigured to a lower priority (high->low->off).

3 Experimental Results

The proposed architecture has been tested using a sequence of 270 images from the KITTI dataset in a Core i5 CPU at 3.10 GHz. Soft faulty data for vision and reference sensors have been simulated, introducing a displacement in the calibration matrix from a camera and from the Velodyne (miscalibration)respectively. In addition, hard fault in a vision sensor was simulated fixing the output of a camera on a constant value (lost signal).

The SVM models were trained using a subset of 25 representative images from the 270 testing set. For creating the MF model, 261214 vectors (130607 positives and 130607 negatives) with a 'high' weight value for all the sensors were trained offline in 23.6 minutes. The FDD model for the vision sensors were trained in 0.06 seconds using 500 vectors (250 positives and 250 negatives) each, while the training of the FDD model for the Velodyne sensor lasted 0.08 seconds with 1219 vectors (546 positives and 673 negatives)

Figure 5 shows the output of the SVM algorithm trained with the FDD model for camera 1. When a no faulty data is processed (red) it responds with sporadic positive values(false positives); however, these responses are not persistent. On the other hand, faulty data (blue) produce persistent outputs(N consecutive images), which generate true positivefaults. In the case of a hard fault the output produces a positive value for a longer time, resulting in a faster fault detection response.

The output of the SVM algorithm trained with the FDD model for Velodyne is shown in Figure 6. Since the Velodyne sensor is the reference of the fusion, it creates strong discrepancies in every LF module, reacting in a similar way as a hard fault in any camera.

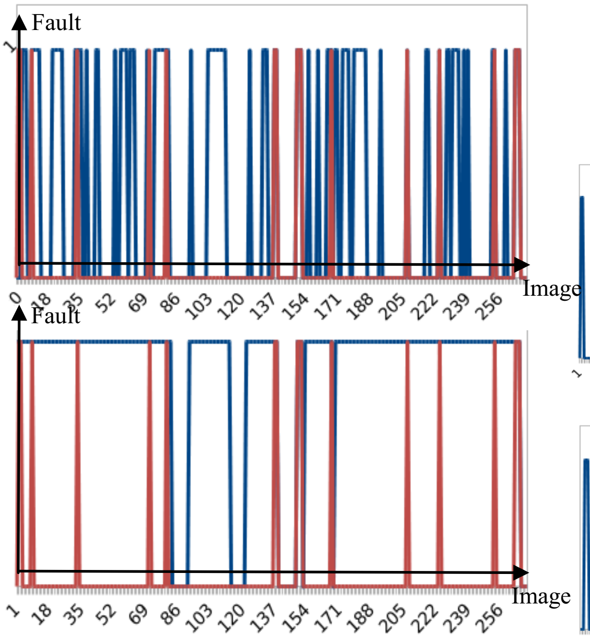


Figure. 5.SVM result for camera1 top) Soft Fault.bottom) Hard Fault.

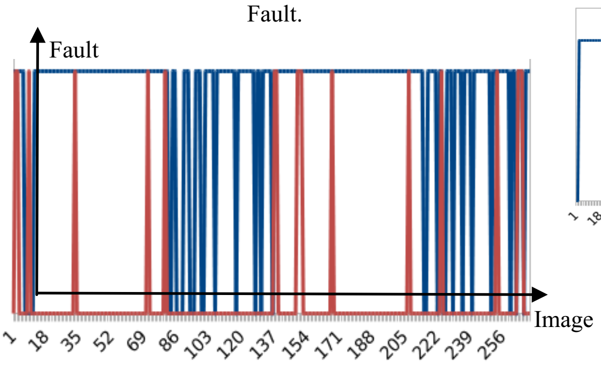


Figure. 6.SVM result for soft fault in Velodyne.

Figure 7 shows the output of the FDD module for a soft and hard fault in camera 1 (blue, red) and for a soft fault in the Velodyne sensor (green) from the tests in Figures 5 and 6. The value N forconsecutive imageswas set to 5. Thus, the respective sensor was reconfigured to a lower priority every time that a response resulted positive for 5 consecutive images.

A translation value has been introduced in the calibration matrix in order to simulate miscalibration in a camera. The translation represents the displacement of the detected objects located in the frontal area of the vehicle at medium range (5 -30 meters). The translation value has been altered to represent displacements from 16 to 51 cm and FDD results have being recorded as shown in Figure 8 and 9.

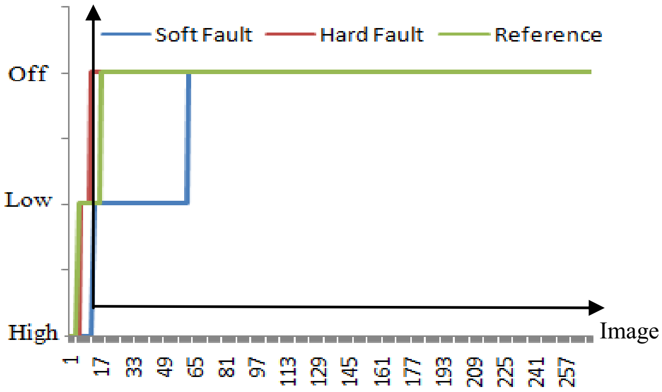
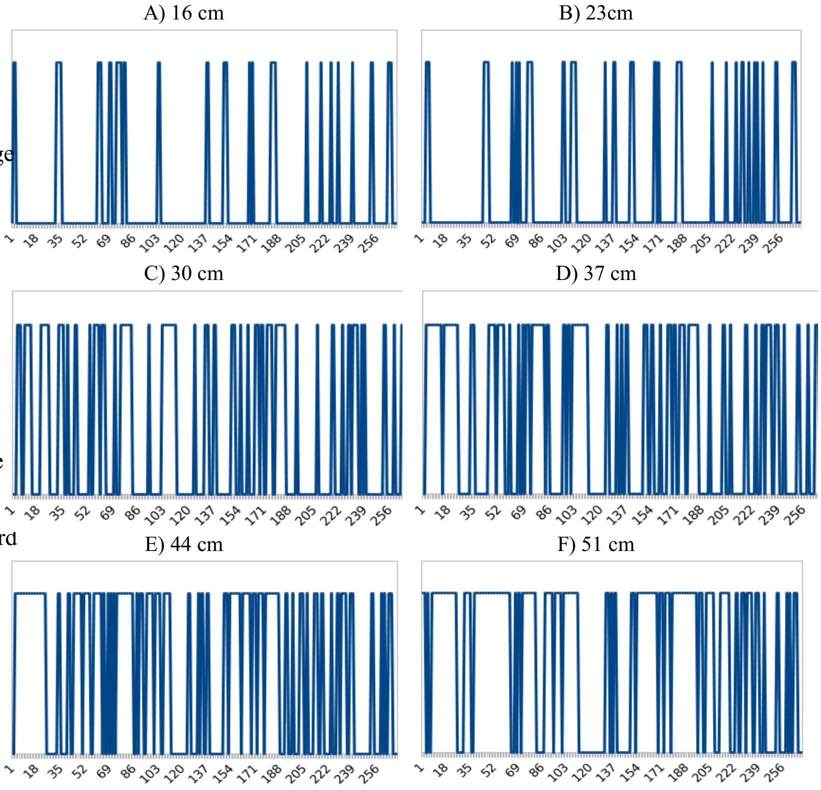


Figure. 7. Fault output from FDD.



A) 16 cm

B) 23cm

C) 30 cm

D) 37 cm

E) 44 cm

F) 51 cm

Figure. 8.SVM result for different soft fault displacements.

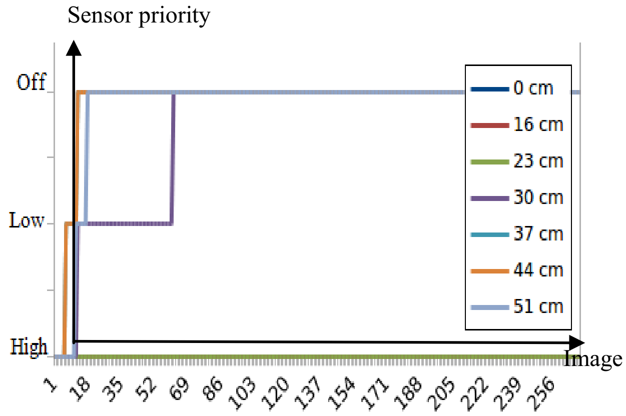


Fig. 9. Fault output from FDD for different soft fault displacements.

4 Conclusions

A federated data fusion architecture in the context of the JDL fusion model has been proposed in order to provide fault tolerance to one sensor of an autonomous vehicle’s perception system. This architecture integrates the process refinement layer to the fusion process, reconfiguring the participation of the sensors in the perception layer.

FDD module has successfully detected faults when displacements of 30 cm of higher were introduced in a camera. Smaller displacements were not detected; however, those displacements errors were corrected in the MF module using the outputs of the other sensors. Since a FDD model was developed for each sensor, no fault diagnosis was needed. However, this solution is not practical for large amounts of sensors. Thus, a single FDD model for all sensors is being developing.

Future work is being carried out to evaluate the system with different Velodyne soft faults. In addition MF is being training with ‘low’ and ‘off’ weight values in order to compensate for large soft sensor faultsSince the nature of the proposed vision based OD algorithm is focused on mobile obstacles,many false positives detections are introduced by static objects showing high discrepancies between vision OD and Velodyne OD. Thus, a new training vectorthat groups the values of the discrepanciesinto static and dynamic features will be tested.

Acknowledgement

This work is supported by the National Secretary of Superior Education, Science, Technology & Innovation of Ecuador (SENESCYT) through its scholarships program and the Escuela Superior Politecnica del Litoral. The authors would like to thank Prof. Dr. Christoph Stiller and the Institut für Mess- und Regelungstechnik of the Karlsruher Institut für Technologie (KIT) for providing access to the KITTI dataset.

References

1. R. Isermann, Fault-Diagnosis Applications: Model-Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems: Springer Berlin Heidelberg, 2011.
2. M. Muenchhof, "Comparison of change detection methods for a residual of a hydraulic servo-axis," pp. 1854-1854, 2005.
3. C. W. Chan, et al., "Application of Fully Decoupled Parity Equation in Fault Detection and Identification of DC Motors," Industrial Electronics, IEEE Transactions on, vol. 53, pp. 1277-1284, 2006.
4. T. Escobet and L. Trave-Massuyes, " Parameter estimation methods for fault detection and isolation," Bridge Workshop Notes, 2001.
5. M. Hilbert, et al., "Observer Based Condition Monitoring of the Generator Temperature Integrated in the Wind Turbine Controller," EWEA 2013 Scientific Proceedings : Vienna, 4 -7 February 2013, pp. 189-193, 2013.
6. G. Heredia and A. Ollero, "Sensor fault detection in small autonomous helicopters using observer/Kalman filter identification," in Mechatronics, 2009. ICM 2009. IEEE International Conference on, 2009, pp. 1-6.
7. N. Meskin and K. Khorasani, Fault detection and isolation : multi-vehicle unmanned systems. New York: Springer, 2011.
8. H. A. Aldridge, "Robot position sensor fault tolerance," Ph.D. 9713717, Carnegie Mellon University, United States -- Pennsylvania, 1996.
9. P. J. Lawrence, Jr. and M. P. Berarducci, "Comparison of federated and centralized Kalman filters with fault detection considerations," in Position Location and Navigation Symposium, 1994., IEEE, 1994, pp. 703-710.
10. N. A. Carlson, "Federated filter for fault-tolerant integrated navigation systems," in Position Location and Navigation Symposium, 1988. Record. Navigation into the 21st Century. IEEE PLANS '88., IEEE, 1988, pp. 110-119.
11. A. Edelmayer and M. Miranda, "Federated filtering for fault tolerant estimation and sensor redundancy management in coupled dynamics distributed systems," in Control & Automation, 2007. MED '07. Mediterranean Conference on, 2007, pp. 1-6.
12. T. Xu, et al., "A multi-sensor data fusion navigation system for an unmanned surface vehicle," Proceedings of the Institution of Mechanical Engineers, vol. 221, pp. 167-175,177-186, 2007.
13. L. Xu and Z. Weigong, "An Adaptive Fault-Tolerant Multisensor Navigation Strategy for Automated Vehicles," Vehicular Technology, IEEE Transactions on, vol. 59, pp. 2815-2829, 2010.
14. D. Fengyang, et al., "Study on Fault-tolerant Filter Algorithm for Integrated Navigation System," in Mechatronics and Automation, 2007. ICMA 2007. International Conference on, 2007, pp. 2419-2423.
15. F. E. White, "Data Fusion Lexicon " JOINT DIRECTORS OF LABS WASHINGTON DC. 1991.
16. A. N. Steinberg, et al., "Revisions to the JDL data fusion model," Sensor Fusion: Architectures, Algorithms, and Applications III, vol. 3719, pp. 430--441, 1999.
17. A. Polychronopoulos and A. Amditis, "Revisiting JDL model for automotive safety applications: the PF2 functional model," in Information Fusion, 2006 9th International Conference on, 2006, pp. 1-7.
18. Realpe M., et al., "Towards Fault Tolerant Perception for autonomous vehicles: Local Fusion," presented at the 7th IEEE International Conference on Robotics, Automation and Mechatronics (RAM), Angkor Wat – Cambodia, 2015.

19. A. Geiger, et al., "Are we ready for autonomous driving? The KITTI vision benchmark suite," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 3354-3361.
20. A. Geiger, et al., "Vision meets robotics: The KITTI dataset," The International Journal of Robotics Research, vol. 32, pp. 1231-1237, September 1, 2013 2013.
21. J. Fritsch, et al., "A new performance measure and evaluation benchmark for road detection algorithms," in Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on, 2013, pp. 1693-1700.
22. T. Joachims, "Making large-scale support vector machine learning practical," in Advances in kernel methods, ed: MIT Press, 1999, pp. 169-184.
23. V. Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models: MIT Press, 2001.