

Multi-sensor Fusion Module in a Fault Tolerant Perception System for Autonomous Vehicles

Miguel Realpe^{1,2}

¹Intelligent Control Systems Laboratory, Griffith University. Brisbane, Australia

²Escuela Superior Politécnica del Litoral, ESPOL, CIDIS - FIEC, Campus Gustavo Galindo Km 30.5 Vía Perimetral, Guayaquil, Ecuador

Email: mrealpe@fiec.espol.edu.ec

Boris X. Vintimilla² and Ljubo Vlacic¹

¹Intelligent Control Systems Laboratory, Griffith University. Brisbane, Australia

²Escuela Superior Politécnica del Litoral, ESPOL, CIDIS - FIEC, Campus Gustavo Galindo Km 30.5 Vía Perimetral, Guayaquil, Ecuador

Email: bvintim@fiec.espol.edu.ec, l.vlacic@griffith.edu.au

Abstract—Driverless vehicles are currently being tested on public roads in order to examine their ability to perform in a safe and reliable way in real world situations. However, the long-term reliable operation of a vehicle’s diverse sensors and the effects of potential sensor faults in the vehicle system have not been tested yet. This paper is proposing a sensor fusion architecture that minimizes the influence of a sensor fault. Experimental results are presented simulating faults by introducing displacements in the sensor information from the KITTI dataset.

Index Terms—Fault Tolerance, Data Fusion, Multi-sensor Fusion, Autonomous Vehicles, Perception System.

I. INTRODUCTION

The deployment of driverless vehicles has attracted a lot of attention from researchers and the automobile industry in the past decade, given that they would bring fundamental improvement to vehicle safety, traffic accident rates and the environmental impact of automobiles.

A key issue for developing safe driverless vehicles is accurate and reliable perception because the self-driving algorithm strongly relies on this information. The objective of perception systems in driverless vehicles is to provide a description of the environment around the vehicle, and obstacles in particular.

Driverless vehicles are currently being tested on public roads in order to examine their ability to perform in a safe and reliable way in real world situations. However, they have limited exposure to various traffic scenarios and events so far [1]. In addition, the long-term behaviour of vehicle’s diverse sensors has not been tested. Finally, fault-tolerant perception architectures are still at the developmental stage.

From the reliability perspective of the vehicle perception system, any fact or event that negatively

affects the capability of the perception system to acquire and process correct information from the sensors in order to detect obstacles around the vehicle is a threat. Threats can be classified as faults, errors, or failures; their relationship is illustrated by the fault-failure chain shown in Figure 1. A fault can be either a hardware defect or a software imperfection. When activated during system operation, a fault leads to an error. A failure occurs if an error is not detected, resulting in the vehicle behaviour that is inconsistent with its specification.

Sensor faults can be divided into two general types, hard and soft faults. Hard faults are presented in a stepwise form when sensor data changes abruptly from its normal state to a faulty one, while soft faults are slow degradations in the sensor data through time. Soft faults are more difficult to detect and eliminate because it takes time before the sensor data leaves a limit of confidence. Meanwhile, the faulty information (unless it is detected) may be still used by the system and, thus, may be added to the set of correct data.

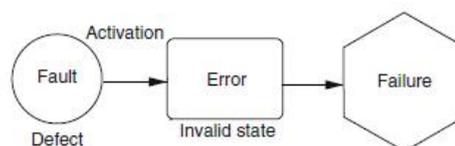


Figure 1. Threat fault – failure chain

A variety of sensors has so far been mounted on driverless vehicles. However, some sensors, such as Lidars and cameras, have gained notoriety in urban traffic vehicle applications.

The Velodyne Lidar sensor [2] provides a file that contains correction factors for the proper alignment of the point cloud information gathered for its lasers in order to correct systematic errors (biases) in sensor readings. However, in practice those parameters are not very accurate [3]. For instance, points with uncertainties in the

order of 30 cm are reported in [4], even after applying correction factors supplied by Velodyne in addition to distance offset calibrated using the readings from another reference Lidar sensor.

This paper is proposing a sensor fusion architecture that minimizes the influence of a sensor faults by combining data from a federated fusion structure with the sensors weight feedback data provided in real-time by the Fault Detection and Diagnosis module, using a support vector machine (SVM) algorithm. Experimental results are presented simulating faults by introducing displacements in the sensors data. This paper is organised as follows: The proposed model is presented in Section 2. Experimental results are provided in Section 3. Finally, conclusions are given in Section 4.

II. MODEL DESCRIPTION

The implementation of the perception system has been done based on the sensors available in the KITTI dataset [5-7], which includes a Velodyne sensor and two pairs of stereo vision cameras. The general perception layer suggested to fuse sensor data from the KITTI dataset which is shown in Figure 2, provides information to the Decision Application Layer (DAL) and has been described in previous works [8, 9]. The system is divided into different modules: Object Detection (OD), Local Fusion (LF), Master Fusion (MF) and Fault Detection and Diagnosis (FDD).

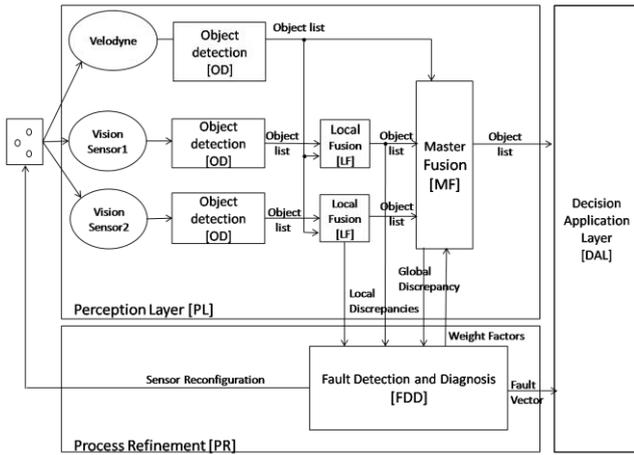


Figure 2. Fault Tolerant perception system for KITTI dataset.

A. Fault Tolerance

The FDD module recognises changes in the discrepancies values from the MF and LF modules and provides sensors weight feedback. The sensor weights are reconfigured from a high level of influence to a low level when a fault is detected. If the fault remains, the sensor weight is changed to off. Figure 3 (adapted from [9]) shows the outputs of FDD executed in a sequence of images when a fault is present in a vision sensor (blue) and when it is present in the Velodyne sensor (green). The sensor weight starts with a value of zero, which is later interpreted by MF as high. When the fault is

detected, FDD output changes to 1; whose value is coded as low in MF. Finally, the output is set on 2, which is read in MF as off. In addition, FDD sends the weights of the sensors to the DAL module, which interprets them as a fault vector.

Figure 4 shows the state diagram of the Fault Tolerant Perception System from Figure 2. The transition from a correct state (C) to a tolerated error state (T) is constrained by the outputs of FDD. When the system is running without any fault it is catalogued as being in a correct state C. On the other hand, the Sensor Weights Vector determines when a fault is detected, changing the influence of a faulty data in the sensor fusion process and modifying the state of MF to a tolerated error state T in the next execution cycle. In addition, FDD reconfigures the sensors turning them off in concordance with the fault vector. Moreover, when the system is running in a tolerated error state a recovery action, such as resetting or recalibrating the sensor, should be executed by DAL, otherwise the error may persist and propagate causing a failure in the system.

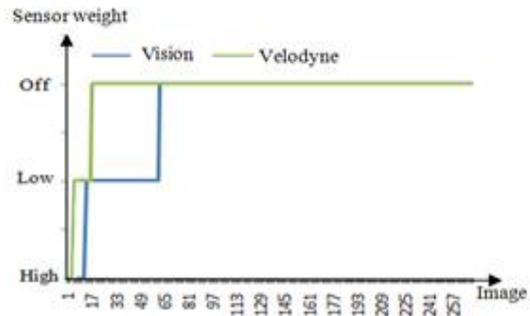


Figure 3. Weight output from FDD for soft faults of vision sensor and Velodyne sensor (adapted from [9]).

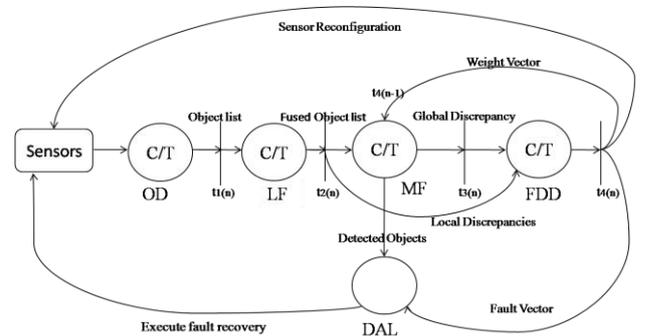


Figure 4. General state diagram of the Fault Tolerant Perception System.

An example of the system's states being executed with a faulty sensor is shown in Figure 5 (Detected Objects signal from MF to DAL is not shown for clarity purposes). The system starts in a correct state and all sensors weights are set on high (HHH) at time $t=0$, and it keeps running in state C until the fault is detected at time $t=n$. When FDD detects the faulty data from the sensor, its weight is set on low (LHH). Accordingly, in the next

execution cycle ($t=n+1$) FDD and DAL change the system state to a tolerated error state, while the faulty data influence is minimised in the MF module. As the fault remains taking place, at time $t=m$ the sensor weight is changed to off (OHH). At the next cycle ($t=m+1$), the FDD module reconfigures the sensor turning off its signal. If there is a second sensor fault at this execution time, the FDD changes its weight to low (OLH), resulting in a failure of the system.

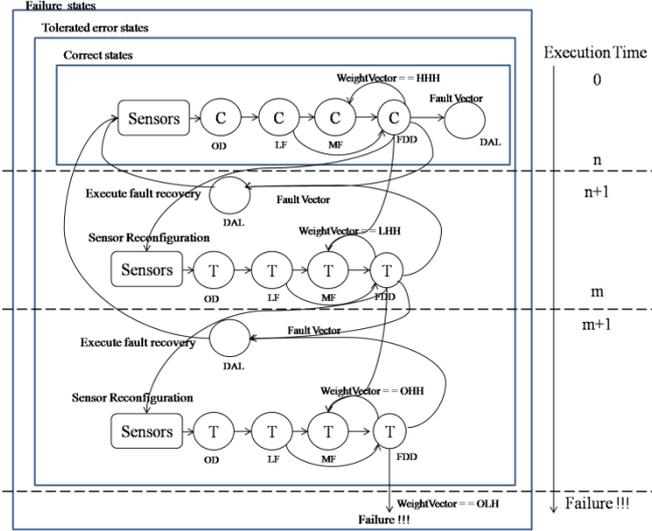


Figure 5. System fault state-space.

B. Sensor Fusion

A federated data fusion architecture based on the JDL model [10] was implemented in order to provide multi-sensor fusion in a fault tolerant context. This architecture integrates the FDD to the fusion process, reconfiguring the participation of the sensors in the perception layer. Accordingly, MF uses the weight vector values from FDD to validate data fusion from LF.

MF combines data from sensors in order to create a single list of obstacles. Detected objects from the MF inputs are considered 2D blobs described by their position, width and height expressed in the coordinates system of vision sensor 1. In order to convey objects from LF related to vision sensor 2 to the coordinate reference system of vision sensor 1, the centre points of the blobs from LF2 are individually aligned with those from LF1 using the closest neighbour [11] based on their Euclidean distance. On the other hand, each point (X, Y, Z) in the objects from the Velodyne OD is transformed into the point (x,y) in the coordinate system of vision sensor 1, using the 3×4 projection matrix P as follows:

$$\mathbf{x} = \frac{u}{w}, \quad y = \frac{v}{w} \quad (1)$$

$$[u \ v \ w]' = P \cdot [X \ Y \ Z \ 1]' \quad (2)$$

Where P is composed of the 3×3 rotation matrix R and the 3×1 translation matrix T from the Velodyne calibration.

$$P = [R \ | \ T] \quad (3)$$

Once all the objects are expressed in the same coordinate system, they are clustered based on the amount of their overlapping areas, using the rectangle equivalence criteria in OpenCV [12, 13], that combines rectangles with similar sizes and locations into candidate objects.

Then, the weight of each sensor provided by FDD and patterns in the objects' pixels sensor are used to validate the pixels of the candidate objects through a SVM classification model. A general pattern classification problem is posed as follows [14]: Given a training sample S , consisting of n independent identically distributed observations of the form.

$$S = \{(x_i, y_i) \mid x_i \in \mathbb{R}^p, y_i \in \{-1, +1\}\}_{i=0}^n \quad (4)$$

Where each x_i is a feature vector of length p and y_i represents the class label for the data point x_i . The problem consist of finding a classifier with the decision function $f(x): x \rightarrow \{-1, +1\}$ based on S that classifies new points as accurately as possible. SVM [15] seeks the hyperplane that divides the points having $y_i=1$ from those having $y_i=-1$, in addition to exhibit the largest distance to the nearest points of each class (maximum margin).

MF defines a training set S where each observation is composed by (x_i, y_i) , as shown in table 1. The value of the vector $x_i \in \mathbb{R}^9$ is obtained from the outputs of each LF module and the OD from the Velodyne sensor, while y_i is defined manually in all the pixels of a group of images from the reference vision sensor. The first triplet represents the presence or absence of the pixel in a detected object from the reference sensor and the local fusion modules. It is given by an image where objects are black (0) and the rest of the pixels are white (255). The second triplet denotes the distance fields that show the distance of the corresponding pixel to the closest detected object. Distance fields values range from 0 to 255 representing the closest and furthest distances respectively. The last triplet is given by the FDD output.

TABLE I. TRAINING SET OBSERVATION COMPOSITION (x_i, y_i) .

x_i	Value
Reference Sensor	True (0), False (255)
Local Fusion 1	True (0), False (255)
Local Fusion 2	True (0), False (255)
Reference distance field	0-255
Local distance field 1	0-255
Local distance field 2	0-255
Weight reference	high (0), low(1), off (2)
Weight vision 1	high (0), low(1), off (2)
Weight vision 2	high (0), low(1), off (2)
y_i	Value
Pixel validation	-1, 1

After pixels of the candidate objects are validated with SVM, multiple object tracking is performed in order to provide information about the future position of objects. Prediction of their future location is done by applying the Kalman filter to every blob representing an object. Then, the Hungarian algorithm [16] connects all the predictions

to previous tracks. It also determines the tracks that were missing and which objects should begin a new track. In addition, the location variation in time of the tracked objects is used to determine if they are static or dynamic obstacles.

III. EXPERIMENTAL RESULTS

Experiments have been done with the intention of analysing the effects that individual faults in vision sensor 1 and the Velodyne sensor may have in the MF module when the system alters from correct state C to tolerated error state T. The proposed architecture has been tested using a sequence of 261 images from the KITTI dataset in a Core i5 CPU at 3.10 GHz.

In order to simulate soft faults in the reference sensor, displacements of $(100 * A)$ cm were introduced in the cloud of points given by the Velodyne sensor, using the matrix $E = [R_E \ T_E]$, where $T_E = [0 \ A \ 0]$ and R_E is the unit matrix of size 3. Similarly, a translation value in pixels was experimentally defined to simulate image faults by comparing 2D projections of the Velodyne sensor before and after applying matrix E.

To create the SVM model for MF, 505620 vectors (227170 positives and 278450 negatives) were trained offline with 'high', 'low' and 'off' weight values for all the sensors. Figure 6 shows an example of the outputs from LF and MF when data without faults is processed, which was later used as a reference to compare changes in the modules outputs when error data was simulated. In the figure, the static objects are represented with red colour, while dynamic objects are shown in purple.



Figure 6. Top) LF1, Middle) LF2, Bottom) MF results for a set of correct data

For the first experiment a displacement to the right by 30 cm in the objects from vision sensor 1 was introduced, which as consequence produced the creation of new objects and the displacement of the detected objects in LF1 (Figure 7) in relation with the LF1 results for a set of

correct data (top image in Figure 6). Then MF was run in a correct state with a weight value of high for all the sensors. Eventually, FDD changes the weight for vision sensor 1 from high to low and then off. Once the weight is set to off, the DAL module should modify the sensors so no more data from that faulty sensor is updated. In addition, when FDD reduces the priority of a sensor, the system enters a tolerated error state, where the DAL module should evaluate if a recovery action needs to be executed. Since the nature of the proposed vision based OD algorithm is focused on moving obstacles, the data that is fused from vision sensors and Velodyne is largely related to dynamic objects. Thus, analysis of results will be centred on dynamic objects.

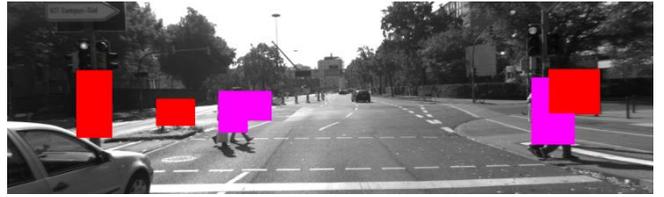


Figure 7. LF1 with fault simulation in vision sensor 1.

Figure 8 shows the result of MF when a faulty data from vision sensor 1 is simulated. In the top image the stronger influence of the faulty data can be appreciated when the sensor weight is set on high, displacing the objects slightly to the right in relation with the MF results for a set of correct data (bottom image in Figure 6). In addition, the faulty data adds more false positives pixels. Nevertheless, once FDD changes the weight for the faulty vision sensor to low (Figure 8 middle) and off (Figure 8 bottom) the false positives pixels are reduced.

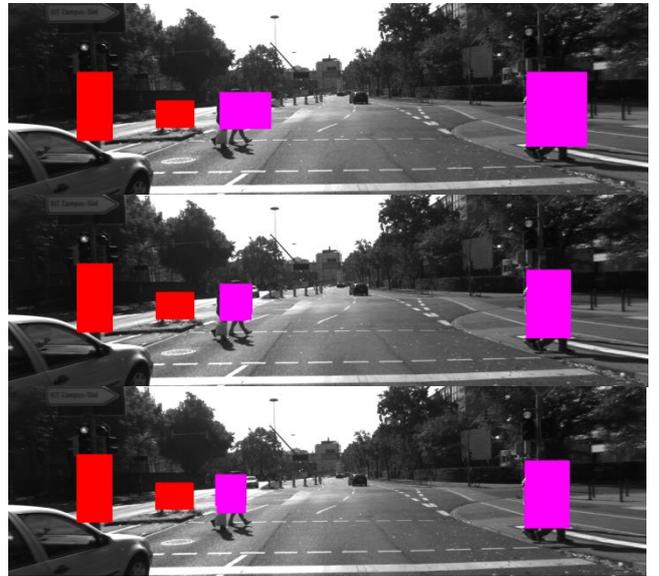


Figure 8. MF with faulty data from vision sensor and weight set on: Top) High, Middle) Low, Bottom) Off.

Nonetheless, the number of detected objects and the number of pixels positively classified as part of an object

are similar for all the cases (with and without fault) due to the redundancy of data with vision sensor 2. This similarity can easily be appreciated in Figure 9, which shows the percentage of pixels from the detected objects that are positively classified by MF as dynamic objects in a sequence of 261 images, where the blue line represents results using a set of correct data while red, yellow and green represent results using a faulty data from vision sensor 1 with weights values of high, low and off respectively.

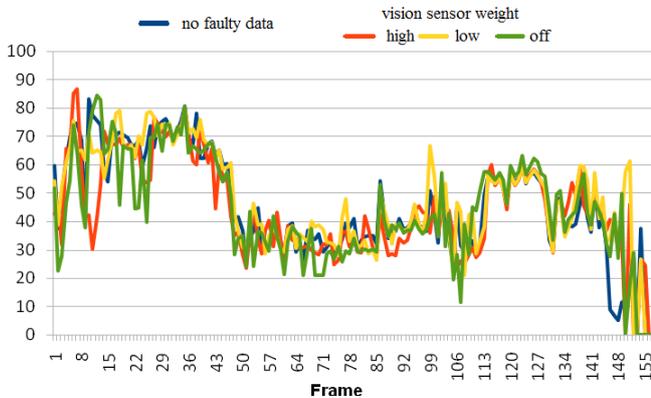


Figure 9. Positive detection of MF with faulty data from vision sensor (dynamic objects)

Figure 10 depicts the percentage of pixels from the detected dynamic objects that are false positives in a sequence of 261 images, where the blue line represents results using a set of correct data while red, yellow and green represent results using faulty data from vision sensor 1 with weights values of high, low and off respectively. The number of false positives increases when the system is running in correct state C with a faulty sensor's weight set on high. However, false positives are reduced by up to 57% with an average of 7% when the faulty sensor's weight changes to low and by up to 59% with an average of 11% when the faulty sensor's weight is modified to off.

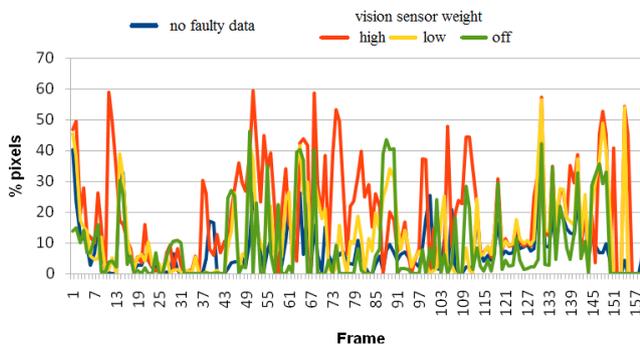


Figure 10. False positives detection of MF with faulty data from vision sensor (dynamic objects)

For the second experiment a displacement to the left (in the vision sensor 1 coordinate system) of 30 cm in the objects from the Velodyne sensor was introduced, which produced displacements in the detected objects in LF1 and LF2 (Figure 11) in relation with the LF1 and LF2

results for a set of correct data (top and middle image in Figure 6). Then, MF was run in a correct state with a weight value of high for all the sensors. Eventually, FDD changes the weight for Velodyne from high to low and then to off.

Figure 12 shows the result of MF when a faulty Velodyne is simulated. The percentage of pixels from detected dynamic objects is similar when there is no presence of faulty data and when the faulty Velodyne is detected and corrected to low weight (Figure 13). When the Velodyne sensor is turned off, MF relies only on the information from the vision sensors, as a consequence the percentage of pixels from detected dynamic objects increases in relation to the total of pixels of detected objects because visual OD is focused on dynamic objects.

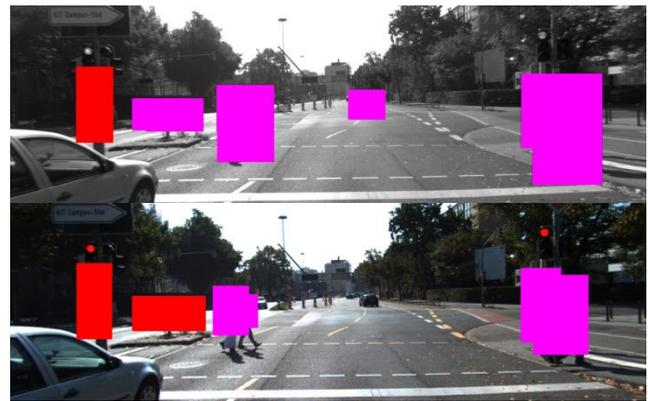


Figure 11. Top) LF1, Bottom) LF2, with fault simulation in Velodyne sensor.

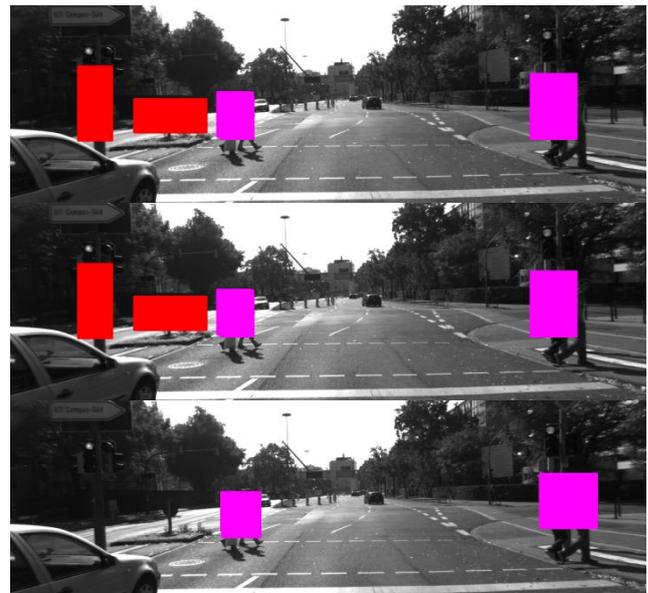


Figure 12. MF with faulty data from Velodyne weight set on: Top) High, Middle) Low, Bottom) Off

Figure 14 depicts the percentage of pixels from the detected dynamic objects that are false positives. The blue line corresponds to results using a set of correct data while red, yellow and green represent results using faulty

data from Velodyne with weights values of high, low and off respectively. When the system is running in correct state with a faulty Velodyne whose weight is set on high (red line) the number of false positives increases. Nevertheless, false positives are reduced by up to 39% with an average of 3% when the weight of the faulty Velodyne is modified to low (yellow), and by up to 55% with an average of 10% when the weight of the faulty Velodyne changes to off (green line).

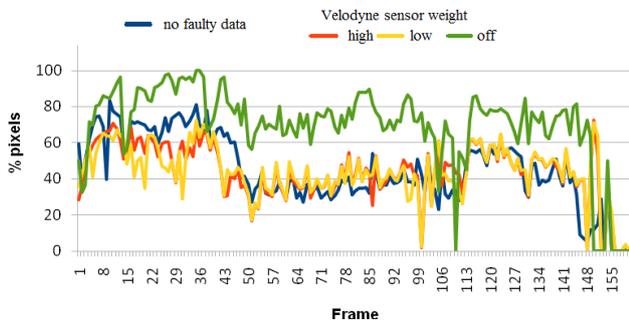


Figure 13. Positive detection of MF with faulty data from Velodyne (dynamic objects)

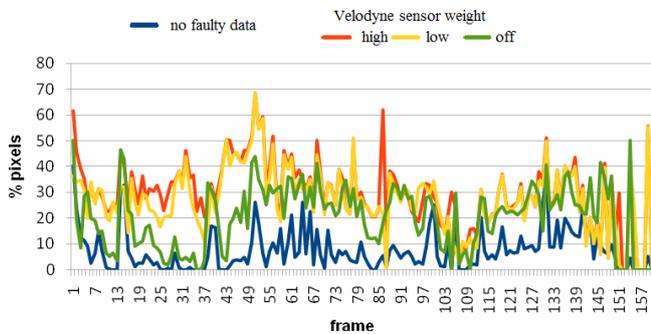


Figure 14. False positives detection of MF with faulty data from Velodyne (dynamic objects)

IV. CONCLUSIONS

One objective of the fusion process is to reduce the influence of faulty data, but it also tends to mask errors in sensor, which may persist in the system and even increase over time. This paper presents a sensor fusion architecture that minimizes the influence of a faulty sensor, in addition to the early detection of faults by combining data from Sensor Fusion with Fault Detection and Diagnosis.

When a faulty sensor was simulated, MF maintained the percentage of positive detection of pixels even when the system was running in correct state C. Thus, the impact of the sensors weights was minimal in the positive detection of objects. However, a reduction in the false positive detection was observed when FDD changed the weights of a faulty sensor.

False positives are reduced by an average of 3% and 10% when the weight of faulty data from Velodyne is modified to low and off respectively. Similarly, false positives are reduced by an average of 7% and 11% when faulty vision sensor's weight is modified to low and off respectively.

ACKNOWLEDGMENT

A grant from "SENESCYT - Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación de la República del Ecuador" and the ESPOL University supported Miguel Realpe.

REFERENCES

- [1] M. S. Brandon Schoettle, "A Preliminary Analysis of Real-World Crashes Involving Self-Driving Vehicles," 2015.
- [2] Velodyne. *HDL-64E*. Available: <http://velodynelidar.com/lidar/hdlproducts/hdl64e.aspx>
- [3] N. Muhammad and S. Lacroix, "Calibration of a rotating multi-beam lidar," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 5648-5653.
- [4] J. Bohren, et al., "Little Ben: The Ben Franklin Racing Team's entry in the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, pp. 598-614, 2008.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] M. Realpe, et al., "Towards fault tolerant perception for autonomous vehicles: Local fusion," in *Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2015 IEEE 7th International Conference on*, 2015, pp. 253-258.
- [9] M. Realpe, et al., "Sensor Fault Detection and Diagnosis for autonomous vehicles," *MATEC Web of Conferences*, vol. 30, p. 04003, 2015.
- [10] 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.
- [11] S. Arya, et al., "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *J. ACM*, vol. 45, pp. 891-923, 1998.
- [12] G. Bradski, "The opencv library," *Doctor Dobbs Journal*, vol. 25, pp. 120-126, 2000.
- [13] K. Pulli, et al., "Real-time computer vision with OpenCV," *Commun. ACM*, vol. 55, pp. 61-69, 2012.
- [14] T. Joachims, "Estimating the Generalization Performance of an SVM Efficiently," presented at the Proceedings of the Seventeenth International Conference on Machine Learning, 2000.
- [15] V. N. Vapnik, *The nature of statistical learning theory*: Springer-Verlag New York, Inc., 1995.
- [16] H. Kuhn, "The Hungarian Method for the Assignment Problem," in *50 Years of Integer Programming 1958-2008*, M. Jünger, et al., Eds., ed: Springer Berlin Heidelberg, 2010, pp. 29-47.



Miguel Realpe received his degree in Computer Science in 2006 at the Escuela Superior Politécnica del Litoral – ESPOL. He is currently working toward the Ph.D. degree with the Intelligent Control Systems Laboratory, Griffith University. Since 2002, he has been with the Centre of Vision and Robotics, ESPOL. Since April 2016, he has been a Lecturer at the Department of Electrical and Computer Science Engineering of the ESPOL, where he teaches Introduction to Robotics. His main research interests include mobile robotics, multi-robot systems and computer vision.



Boris X. Vintimilla received his degree in mechanical engineering in 1995 at the Escuela Superior Politécnica del Litoral – ESPOL, Guayaquil, Ecuador, and his PhD degree in industrial engineering in 2001 at the Polytechnic University of Catalonia, Barcelona, Spain. In May 2001, he joined the Department of Electrical and Computer Science Engineering of the ESPOL as associated professor and in 2008 became a principal professor. Dr Vintimilla has been the director of the Center of Vision and Robotics from 2005 to 2008.

He did his post-doctorate research in the Digital Imaging Research Center at Kingston University, London, UK, from 2008 to 2009. Currently, he is director of the Center of R+I+i for Computer System at ESPOL. His research interests span a broad spectrum within the image processing and analysis, and vision applied to mobile robotics. Dr Vintimilla has been involved in several projects supported by international and national organizations, as result of these researches he has published more than 40 scientific articles and book chapters.



Professor Ljubo Vlacic is with Griffith School of Engineering and is Founding Director of the Intelligent Control Systems Laboratory as well as Founding Director of Mechatronic Engineering Program, Griffith University. Throughout his career he has held a number of leading roles in both industry and academia.

Professor Vlacic's research into co-operative driverless vehicles made news headlines, and was broadcast widely through various media outlets throughout the world. In recognition of his transformational research into, and the world's first demonstration of co-operative driving by driverless vehicles, Professor Vlacic has been awarded several national and international awards.

He serves/has served as an Associate Editor on several editorial advisory boards including the IEEE Transactions on Intelligent Transport Systems and the IEEE – Intelligent Transport Systems Magazine.

He is general Chair of the 2017 Asian Control Conference and was General Chair of the International Federation of Automatic Control - Intelligent Autonomous Vehicles Symposium 2013; General Chair of the Institution of Electrical and Electronic Engineers – Intelligent Vehicles Symposium 2013; and the 2011 – 2012 Chair of Engineers Australia National Committee on Automation, Control and Instrumentation.