# Multisensor Architecture for an Intersection Management System

Gustavo Velasco-Hernández

June 7, 2018

Multisensor Architecture for an Intersection Management System

Gustavo Velasco-Hernández

June 7, 2018

# Abstract

Your abstract goes here... ...

# Acknowledgements

I would like to thank to all people that in some way or another BLABLABLA

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Goals

The main goal for this proposal is to conceive a multisensor architecture for an intersection management system and deploy it in a simulated scenario. In order to achieve this goal, these objectives were set:

- To make a literature review on intelligent transportation systems, focusing on developments about intersection management
- To define system architecture and specifications, and choose a platform which fits requirements for the deployment
- To develop and implement an intersection monitoring system based on laser and video sensors using sensor fusion techniques
- To propose and deploy a method for incident estimation and warning generation
- To set a test and evaluation protocol for the developments made on the choosen platform

## 1.2 Scope

# Chapter 2

# Multisensor Data Fusion and Intersection Management Systems

An intersection is a highly-dynamic scenario that can be monitored using a wide range of sensors. For this reason, an efficient and accurate fusion of the information is needed. This chapter is divided into two sections. In the first section a brief overview of multisensor data fusion is presented, remarking in different architectures proposed in the literature and different algorithms and frameworks used for this task. The second section contains a short review on intelligent transportation systems and intersection management systems, including a description of elements involved in the development of an IMS application. Finally, some projects that include multisensor data for intersection managing applications are presented.

## 2.1 Multisensor Data Fusion

Data fusion, also referred as mutisensor data fusion, information fusion or sensor fusion, has received several definitions from different authors in the literature. For example, Joint Directors of Laboratories defined data fusion as "multi-level, multifaceted process handling the automatic detection, association, correlation, estimation, and combination of data and information from several sources"[65]. Luo refers to multisensor fusion and integration as "synergistic combination of sensory data from multiple sensors to provide more reliable and accurate information"[43] and "to achieve inferences that are not feasible from each individual sensor operating separately"[45]. Elmenreich states that sensor fusion is "the combining of sensory data or data derived from sensory data such that the resulting information is in some sense better than would be possible when these sources were used individually"[24]. In [14] there is a compilation of more definitions of information fusion and the author summarize in his own statement as follows: "Information fusion is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making".

All of previous definitions can be seen as a way to answer these three questions about data fusion:

- What is involved in data fusion?
  Combine, merge or integrate homogeneous or heterogeneous data.

- What is the aim of data fusion?
  Get a better representation of a process or the environment, infer underlying information, improve quality of the data.

- How to apply data fusion?
  Data fusion is a multi-level task, depending of the nature of the sensors, the final application and the environment.

It is clear, now, that multisensor data fusion is a multidisciplinary field, because information in a typical process, flows from sensors to applications, passing through stages of filtering, data enhancement and data extraction. It is for this that knowledge in a wide range of fields are required, e.g. signal processing, machine learning, probability and statistics, etc. Also, it would be pointless to try to define a general method, technique or architecture that fits the requirements of any system, for applying data fusion in it.

### 2.1.1 Data Fusion Architectures and Models

Although there is not a general rule of how to design or implement a sensor fusion system, many authors have proposed some models, architectures and guidelines for this task. Three well-known models are Waterfall model, JDL fusion model and Multisensor integration fusion model.

#### Waterfall model

Harris and Markin in [32] and CITE, proposed a model named Waterfall, in which they describe the fusion process as an information flow through sensing to decision-making. They describe 3 levels of processing with 2 inner stages each ((2.1). The first level is about transform the raw data from sensor to a better representation of the measured phenomena through signal processing and having in mind sensor models and nature of the process itself. The second level objective is to find a meaningful description of the data, reducing its volume while maximising information. This is done using feature extraction and pattern recognition techniques. The third level is the high level of the process in which situation assessment and decision making are performed, based on data available, configuration parameters, database information or human interaction. Finally, a feedback from high-level to low-level (sensor) is done, advising the whole system for re-calibration or reconfiguration.

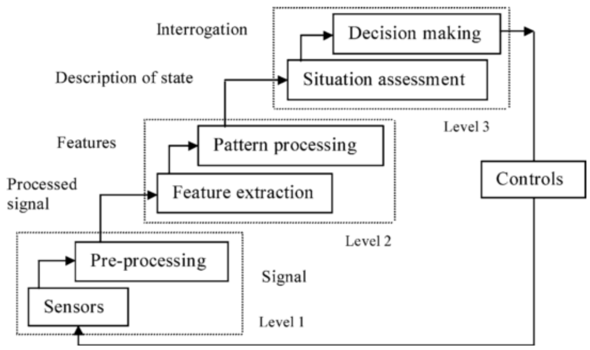


Figure 2.1: Waterfall model (from [26]).

### JDL fusion model

One of the first proposals of fusion architecture, and probably one of the most widely used, is the JDL fusion model, originated from the US Joint Directors of Laboratories and described by Hall and Llinas in [31] and [42]. The JDL fusion model was conceived to aid the developments of military applications and comprises 5 levels of data processing at which data fusion could be done. These levels and a database are connected by a bus (2.2), and are not meant to be execute sequentially and can also be executed concurrently.

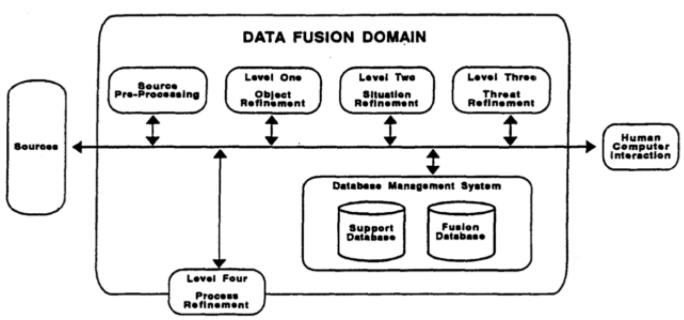


Figure 2.2: JDL fusion model. (from [31]).

The first stage, referred as level-0 is the source preprocessing in which raw data is handled to concentrate the more pertinent data for the current situation. The level-1 is for object refinement, starting with the alignment of the data in a commonly space-time reference frame. Then, performs identification and tracking of objects using different techniques. Situation refinement is at level-2, which takes observed and partially-observed object from level-1 and tries to find a contextual description between them. Level-3, threat refinement, is the level in which results from level-2 are interpreted looking for possible advantages and disadvantages for the system to operate, based on previous knowledge and predictions about executing an action.

#### Multisensor Fusion Integration model

Luo and Kai, in [46, 44], proposed a full integration model for data fusion in which they define a three-level hierarchy for sensor fusion: data-fusion, feature-fusion and decision-fusion. This model separate MFI in five classes, based on Input/Output pair: Data in-data out fusion, data in-feature out fusion, feature in-feature out fusion, feature in-decision out fusion, and decision in-decision out fusion [45] (figure 2.3).

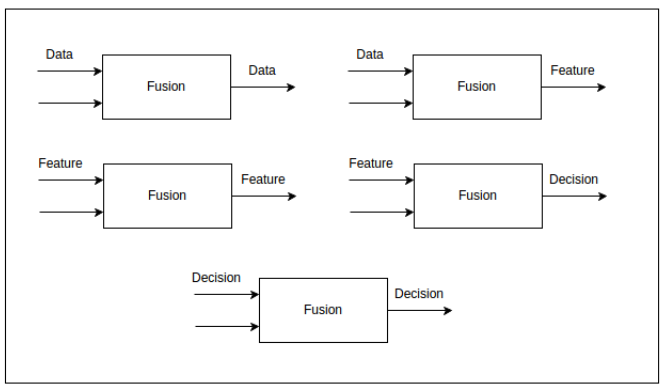


Figure 2.3: Five classes of Multisensor Fusion.

Also, they made a clear distinction between multisensor fusion and multisensor integration, being the former the process in which information provided by a set of sensors is combined in any of the three levels aforementioned, and the latter is how sensor fusion could be integrated in a full system in order to assist in a particular task. As is depicted in figure 2.4, sensor fusion is an element of the whole MFI architecture, which also includes block for sensor managing tasks, like control, selection and registration of sensors, previously to the fusion process. A sensor processing stage and a system controlling module are also included after the sensor fusion stage.

### 2.1.2 Classification of Data Fusion Architectures

Elmenreich in [24] classify fusion models in three categories: Abstract models, generic architectures and rigid architectures. Abstract model are not intended

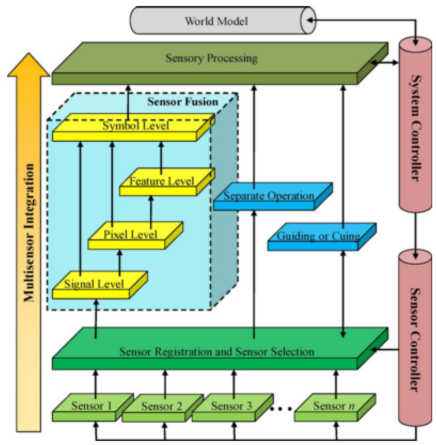


Figure 2.4: Multisensor Fusion Integration architecture (from [45]).

| Category | Data Fusion Model |
|---|---|
| Abstract Model | Waterfall Model [32]<br>Boyd Loop [15] |
| Generic Architecture | JDL Model [65]<br>Multisensor Fusion Integration Model [46]<br>Omnibus model [13] |
| Rigid Architecture | LAAS Architecture [3]<br>DFuse Architecture [39]<br>Time-triggered Model [25] |

Table 2.1: Classification of data fusion models

to show how to implement a sensor fusion system, but to explain which processes are done in it. Generic architectures gives an outline on how a system could be implemented in an application, but do not specify what type of hardware, database or communication system could be used. Finally, rigid architectures, are a good guide for implementation of data fusion in certain applications, at the cost that several design decisions have been already taken, making expensive the migration to another architecture.

In addition to models previously mentioned, there exists more proposal for data fusion architectures in literature, as can be viewed in table 2.1.

### 2.1.3 Algorithms in Data Fusion

Different types of algorithms have been used in implementing data fusion systems, depending on a variety of conditions like, level of fusion, type of the data, nature of environment, etc. Constrains like processing and memory limitations, centralised or distributed schemes, human-interactive or completely autonomous process, also determine which algorithms should be used in fusion process.

Luo in [45] propose a classification for fusion algorithms based on the level of fusion. Low-level fusion refers to the merge of raw data or signals, mid-level fusion refers to the fusion of features and High-level fusion refers to the process of fuse decisions. Khaleghi in [37] describes a classification for fusion algorithms based on challenging problems that arise from the data to be fused, due to the variety of sensors and the nature of the application environment. This classification is not on the algorithms directly, but on the theory or framework in which they originate. Four types of data are enumerated: Imperfect data, correlated data, inconsistent data and disparate data. These two approaches of fusion algorithms classification are summarized in tables 2.2 and 2.3.

| **Low level fusion** | | **Medium level fusion** | **High level fusion** |
|---|---|---|---|
| *Estimation methods* | | *Classification methods* | *Inference methods* |
| Recursive:<br>• Kalman filter<br>• Extended Kalman filter<br>Non-Recursive:<br>• Weighted average<br>• Least squares | Covariance-based:<br>• Cross covariance<br>• Covariance intersection<br>• Covariance union | • Parametric templates<br>• Cluster analysis<br>• K-means clustering<br>• Learning vector quantization<br>• Kohonen feature map<br>• Artificial neural networks<br>• Support vector machines | • Bayesian inference<br>• Particle filters<br>• Dempster-Shafer theory<br>• Expert systems<br>• Fuzzy logic |

Table 2.2: Classification of fusion algorithms based on level of fusion

| **Data Problem** | **Framework** |
|---|---|
| Imperfection | Probabilistic<br>Evidential<br>Fuzzy reasoning<br>Possibilistic<br>Rough set theoretic<br>Hybridization<br>Random set theoretic |
| Correlation | Correlation elimination<br>Correlation presence |
| Inconsistency | Sensor validation<br>Stochastic sensor modeling<br>Prediction<br>Augmented state framework<br>Combination rules<br>Dempsters' rule |
| Disparateness | Dempster-Shafer theoretic framework<br>Human-centered data fusion<br>Hard-soft data fusion |

Table 2.3: Classification of fusion algorithms based on challeging problems in data

## 2.2 Intersection Management Systems

Intelligent Transportation Systems includes a wide range of applications and services transversal to many knowledge areas. For classifying those services, some taxonomies have been proposed like the ones presented in [61, Ch.1] and [66]. From described categories and classes, Advanced Traffic Management Systems have to be considered when an intelligent handling of traffic needs to be deployed.

One of the most desirable scenarios to improve efficiency and safety is an intersection. This because intersections are places where vehicles arrive from different directions at different velocities, increasing the chances for incidents and crashes. Choi [18] states that 40% of reported traffic accidents in the US, were intersection related. Also, in [20], is reported that for Colombia in 2011, most of the accidents in the main cities were at intersections.

Different types of applications and systems are conceived to address these issues. Some tasks performed by those systems are intersection monitoring, vehicles detection, incident warning, collision avoidance, among others. A typical Intersection Management System is composed by three main components: Data source, that could be infrastructure sensors, like inductive loops, range sensors or cameras, and vehicle sensors and traveling data; decision system, which is the core of the whole system, is in charge of analyse and process information provided by infrastructure, vehicles and authorities in order to identify objects, recognise patterns, predict future incidents, control traffic and generate safe decisions and warnings alerts; and finally, is the presentation and displaying of the output of decision system, through infrastructure using dynamic signals, traffic light controlling, or using direct communication with drivers or vehicles through on-board visualisation/notification system. A block diagram of a generic IMS is presented in figure 2.5

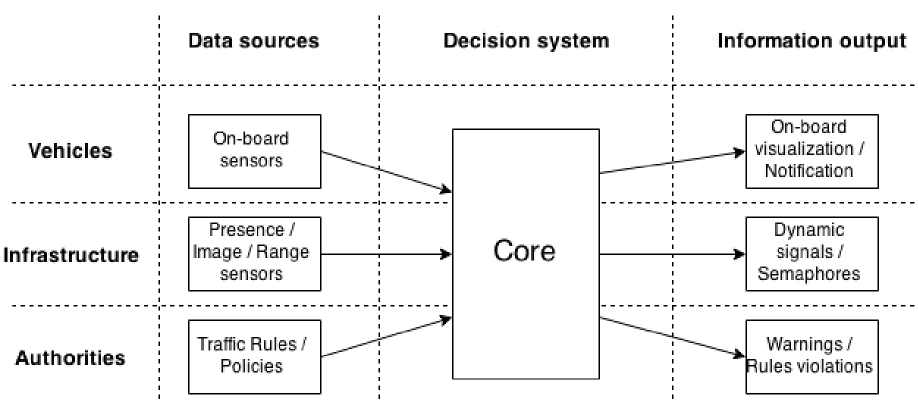


Figure 2.5: Generic block diagram of an Intersection Management System.

### 2.2.1 Components in IMS application

Intersection monitoring is a required task to be done within intelligent transportation systems for high-level applications like traffic analysis, counting and classification of vehicles or pedestrians, event prediction, incident detection and security and surveillance systems. Those applications have to take into account some of the elements depicted in figure 2.5 and developments in IMS have a wide range of approaches and objectives. In order to study IMS applications, five components have been defined, which are present on these applications, and on most cases, more than one component could be involved in the same development. In figure 2.6 a graph is presented, showing aforementioned components and elements within them, and next, a description of each component is given.

#### Application

Application component could be seen as the final objective of the system. Generally, this includes high-level tasks like monitoring, analysis or control. Monitoring or surveillance systems execute actions like recognition, detection and/or tracking of objects in the scene. Other systems analyse the behaviour and interactions between detected objects to recognise path patterns, determine the context of the environment and predict some events of interest. At a higher level there are systems which make decisions based on detection of certain traffic conditions to handle traffic lights, control intersection access, generate warnings to drivers or issue traffic tickets when a rule violation exists.

#### Data Source

The origin of data is considered an independent component because of the variety of possible sources and posterior processing stages. From infrastructure side, data can be captured using a wide range of sensors like inductive loops, lasers, lidars and cameras. Also, monitoring connections to wireless networks. On the other side, data from vehicles is also useful for the system to enhance its representation of the scene and take decisions. This could be low-level data like vehicle state variables, for example, speed, orientation, acceleration, etc., or high-level data like travel information.

#### Target

In an intersection many objects of different kinds interact between them. Pedestrian and vehicles are found at intersections, and latter includes bicycles, motorcycles, cars, vans, buses, trucks and some other types of vehicles. For this reason some applications are designed for a specific element or group of elements. Pedestrian tracking, motorcycles recognition or car counting are examples of targeted applications.

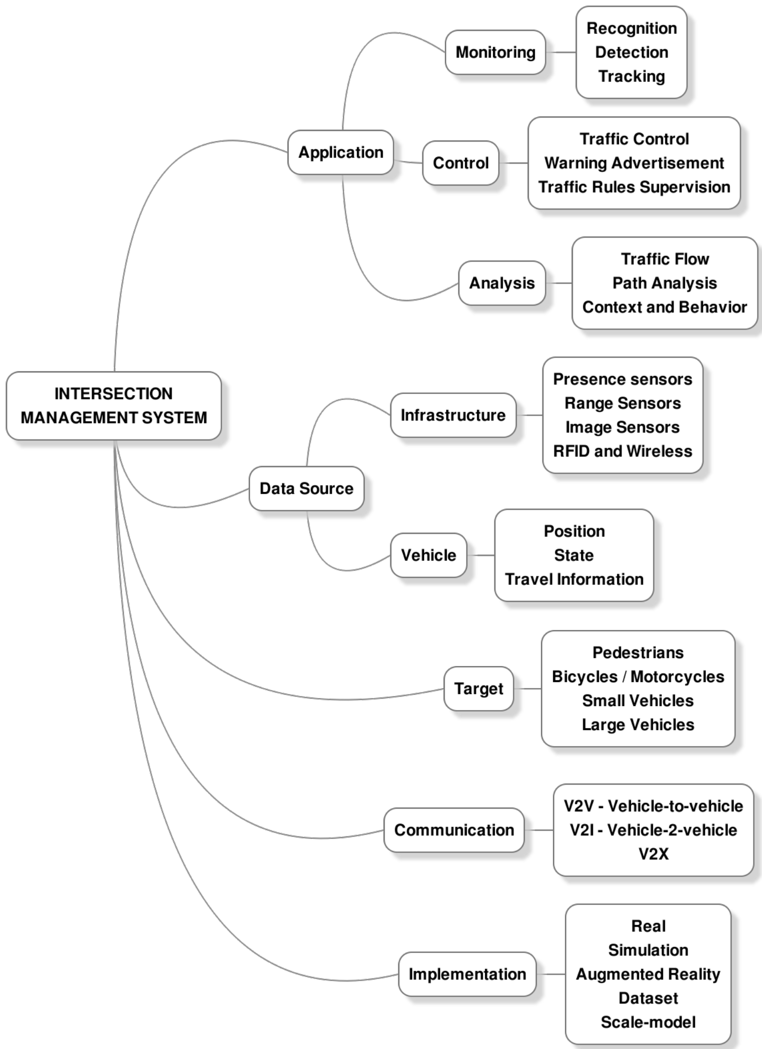


Figure 2.6: Components of an Intersection Management System application.

### Communication

One of the keypoints of ITS is how information technologies and communication advances are included in transportation. The main goal of this is to allow information sharing between vehicles and infrastructure entities. For this reason, 3 communication approaches appear: Vehicle-to-vehicle or Inter-vehicle communication (V2V), vehicle-to-infrastructure communication (V2I) and vehicle to both vehicle and infrastructure (V2X). Several protocols and standars have been proposed for these communication approaches, for example DSRC, WAVE and IEEE 802.11p, but research and development on this component is still active.

#### Implementation

Not all IMS applications are implementable on a real scenario, maybe because is not the scope of the application or because it is in a early stage and it could be validated in other ways. This types of developments are sometimes implemented and evaluated using simulators, making functional prototypes or deploying scale-models. Some other projects use datasets to evaluate new algorithms and data processing techniques, and then compare obtained results with previous works. Augmented reality is also used as a tool for evaluating and validate developments, taking advantage of the interaction of a real system with a simulated/artificial scenario.

### 2.2.2 Developments in Intersection Management Systems

In the table 2.4 is presented a compilation of several developments within IMS field. For each work a brief comment is given, and a remark of how certain components (Data source, communication and application) are related to it. Target and implementation components could be infered from the comment.

| **Developments on Intersection Management Systems** | | | | |
|---|---|---|---|---|
| **Reference** | **Comment** | **Application** | **Data Source** | **Communication** |
| [35, 36] | They show an intersection monitoring system based on a fixed camera. This system is divided in three stages: Background modeling, object tracking and accident detection. They propose an innovative feature for accident detection using HMM. | Accident detection | Camera | N/A |
| [62] | Passive video-based system for monitoring an intersection. They implemented Stauffer's Method for background modeling, PCA for oriented bounding box computation, and Graph-based tracking and motion estimation. Also a simple methods for classification and calibration are described. | Accident detection | Camera | N/A |

| Reference | Comment | Application | Data Source | Communication |
|---|---|---|---|---|
| [63] | They present 4 stages for IMS: background modeling, motion tracking, feature extraction, calibration. They propose a 2-level tracking: Blob tracking as low-level and position using Kalman filter as high-level. | Accident detection / prediction | Camera | N/A |
| [53] | A Full implementation of an IMS in a town in northern Italy. The system claims to be independent of intersection geometry and it is based on a monocular camera. Processing stages of the system and classification approaches are described. | Intersection Monitoring | Camera | N/A |
| [21] | Development of a simulator for an Intersection Collision Warning System. Physical and MAC layers were modeled | Intersection collision warning simulator | N/A | V2V |
| [17] | Traffic monitoring for two context: Left-Turn-Across-Path-Opposite-Direction and Dilemma zone and red light violation | Intersection Monitoring | Radar | N/A |
| [4] | A collision prediction system based on computational geometry is presented. The two main stages are low-level vision system for foreground optimization, with shaking compensation and noise removal, and a collision prediction system based on bounding boxes instead of bounding rectangles. | Collision prediction | Camera | N/A |
| [16] | Analysis of Left-Turn-Across-Path-Opossite-Direction conflict situation. | Intersection Monitoring | Radar; Steering, speed sensors and GPS on-board | V2X |
| [6] | Based on [21], this works includes and improvement, including network layer and driver behaviour model. | Intersection Simulator Architecture | N/A | V2X |

| Reference | Comment | Application | Data Source | Communication |
|---|---|---|---|---|
| [64] | Tracking system based on Kalman filter for tracking and event detection at intersections, capable of detect 4 events: Acceleration, uniform velocity, stop and turns. | Tracking and event detection | Camera | N/A |
| [70] | Object tracking and classification at intersection using a single laser scanner. The system identifies 3 classes: Pedestrians, 2-wheeled vehicle and cars. It is based on points clustering, KL transform and markov chains | Object tracking and classification | Laser | N/A |
| [23] | A system for discriminate an approaching vehicle to an intersection is described. It use a monocular camera and define a number of features for classification. Motion vectors are used too and determine the level of approaching. | Vehicle detection | Camera | N/A |
| [40] | A communication scheme based on nearest to intersection is described. Two zones are defined: nearest zone and quasi-nearest zone. Based on which zone vehicles are, communicaton is performed. Validation of proposal is done using a custom microscopic simulator. | Intersection Communication Scheme | N/A | V2I |
| [22] | A multiagent-based intersection control protocol called autonomous intersection management (AIM) is proposed. Based on a custom simulation, vehicles are modeled and they comunicate with an intersection manager which allows or deny access to intersection. It is possible to set different traffic policies, even for emulate current control approaches lke stop lights and traffic lights. | Intersection Management | On-board state sensors | V2I |

| Reference | Comment | Application | Data Source | Communication |
|---|---|---|---|---|
| [33] | Based on [21] and [6], this simulator architecture emphasize on wireless communications. Now the system allows to compare different communication protocols and traffic configurations. | Intersection Simulator Architecture | N/A | V2V |
| [68] | Based on [70], the system now includes more laser sensor for get a better representation of the scene, solving some occlusion issues. Clustering is used for grouping readings from different sensors belonging to the same object and a tracking approach based on angle of beam, range and time is presented. | Intersection monitoring | Lasers | N/A |
| [56] | Proposal of a communication architecture for vehicles approaching to an intersection. Two zones are defined: Control Channel Zone (CCHZ) and Service Channel Zone (SCHZ). Also, 3 different methods are described for the system to be implmented | Collision avoidance | N/A | V2I |
| [54] | Proposal of a physical-layer protocol for V2I communication at intersections. The system is evaluated in simulation using MATLAB | Collision detection and warning | N/A | V2I |
| [67] | Based on [68], now the system includes a camera for video capture. In this video, data obtained from the lasers-based system are projected, drawing bounding boxes for vehicles and lines for trajectories. | Intersection monitoring | Camera, Lasers | N/A |
| [1] | Monitoring dilemma zone and efectiveness of control policies using cameras and loop detectors on an intersection | Intersection monitoring | Cameras, loop detectors | N/A |

| Reference | Comment | Application | Data Source | Communication |
|---|---|---|---|---|
| [8] | Author presents a system for foreground extraction, vehicle detection and tracking. A novel algorithm is proposed based on image division in traffic zones to perform background removal and vehicle detection. This proposal is compared with traditional MoG approach. | Intersection monitoring | Camera | N/A |
| [47] | A semi-automatic calibration method for a network of lidar sensors is presented using a custom simulation environment. Also, a calibration object were designed based on simulation results. | Intersection Monitoring | Lidars | N/A |
| [55] | An augmented reality implementation of the system proposed in citeDresner2008 is presented. In this work, an autonomous vehicle is deployed in a mixed-reality platform and request to an virtual intersection manager for authorisation to cross and is the manager which decides if it is safe to cross or not, depending on the defined traffic policies and the state of virtual vehicles. | Intersection management | On-board state sensors | V2I |
| [10] | The use of intelligent objects in vehicles is proposed. With sharing vehicle state information and jouney plans, the system aims to better control traffic in intersection using a novel back-off protocol instead of time traffic control system. | Traffic control | On-board state sensors, Journey plans | V2V |
| [9] | A system for traffic flow measurement based on anormalities detection is presented. Multiple cameras are used SIFT and unsupervised SVM-based clustering are performed for trajectories analysis. | Intersection monitoring | Cameras | N/A |

| Reference | Comment | Application | Data Source | Communication |
|---|---|---|---|---|
| [12] | In presented system, wireless magnetic nodes are used to detect presence and to send data to a base station, which determines possible collisions and warns driver through a visualisation system in infrastructure. | Collision Avoidance | Presence sensors | N/A |
| [7] | An intersection control protocol based on V2V communication is proposed. This protocol is in charge of handling other vehicles messages and determines if is it safe or not to cross. They present two policies for managing intersection access: Concurrent Crossing-Intersection Protocol (CC-IP) and Maximum Progression-Intersection Protocol (MP-IP). | Intersection management | N/A | V2V |
| [19] | A microscopic simulator was developed, with a spatial-temporal-based approach for managing the intersection. It is possible to implement different traffic policies for comparison purposes and the system works on crossroads and roundabouts. | Intersection management | On-board state sensors | V2X |
| [69] | A novel approach for detecting moving objects and track them is presented. The focus of this work relies on process merged data from laser sensors ([67]) to get better results at tracking. | Object detection and tracking | Lasers | N/A |
| [34] | A Multiagent-based system is proposed. They define a Vehicle Agent and and Intersection Agent and propose a protocol for controlling intersection access based on timeslots, Also, the present a vehicle motion planning algorithm. Validation and testing was done using SUMO platform | Intersection Management | On-board state sensors | V2I |

| **Reference** | **Comment** | **Application** | **Data Source** | **Communication** |
|---|---|---|---|---|
| [48] | A set of laser scanners is used for detect and track pedestrians. GMM and DBSCAN are proposed for detection stage. For tracking purpose, a random finite set particle filter is used. | Pedestrians detection and tracking | Lasers | N/A |
| [29] | In this work is presented the setting and configuration of a multisensor network, based on 14 lasers, 10 cameras, GPS and V2I unit. Calibration and spatial-temporal alignment is performed. | System Architecture | Cameras, Lasers | V2I |
| [28] | A set of wireless probes are deployed to estimate traffic flow based on bluetooth connections. Then, using Zigbee, data is send to a master node which handle data for forwarding to a server / database for further processing. | Traffic monitoring | Wireless sensor network | N/A |
| [30] | Two modes of Intersection Access Control: V2I mode and V2V mode. Implementation was done using minirobots in a scale model of an intersection. | Intersection access control | Loop detectors and RFID nodes | V2X |
| [50] | A new method for object detection based on lidars data is proposed. After perform background removal, they use DBGridSCAN for 3D clustering and 2D pathway tracking. Then a combination of 2D-3D is done. The proposal is validated using OSPA metric. | Object Detection | Lasers | N/A |
| [5] | On top of AIM ([22]) Semi-AIM protocol is proposed. Assuming that human-driven vehicles to autonomous vehicles transition will be long, this system is intended for semi-autonomous vehicles. Also the author describe the human relation to a semi-autonomous vehicle and analyse how the features of autonomy employed by the car affects the traffic delay. | Intersection Management | On-board state sensors | V2I |

| **Reference** | **Comment** | **Application** | **Data Source** | **Communication** |
|---|---|---|---|---|

Table 2.4: Developments related to Intersection Management Systems.

## 2.3 Sensor Fusion in Intersection Management

Cameras are a wide used sensor for surveillance and monitoring applications because they provide a lot of information for scene understanding, but the transmitting and processing of such information requires high computational resources. For this reason low resolution cameras and low frame rates configurations are commonly deployed. On the other hand, range sensor, such as lasers and lidars, have been recently included in Intersection management systems, as unique sensors or with cameras too. One of the benefits of range sensors is that because of information volume is less than provided by cameras, its processing requirements are lower. Using both cameras and range sensors for intersection management systems makes it possible to take advantages of each type of sensor, and obtain a better representation of the environment.

Although many works based on cameras are found, there are two main developments that make use of range sensors along cameras to implement an IMS. The first one, developed by POSS research group at Peking University[1], is the POSSi project which main objective is monitoring a dynamic environment through fusion of laser and video. In this project, a set of laser scanners are deployed on corners of an intersection and a camera is installed over a pedestrian bridge. They perform a fusion over laser data and after detection and recognition of vehicles and pedstrians, this info is backprojected on video stream.

The second relevant project, Ko-PER project[2], was developed by the Institute of Measurement, Control and Microtechnology at Ulm University under Ko-FAS research initiative. Ko-PER project aims to capture a complete picure of the local traffic environment. In order to do so, a full deployment of infrastructure was made, using 14 laserscanners, 10 cameras, a GPS and a V2I communication unit. Also, they perform a low-level fusion over same type sensor first, and then they merge outputs of each subsystem for improve detection and analysis results.

In the proposed work, there is not a deployment of sensors in an intersection scenario, but datasets from the previous mentioned projects will be used instead. Also, one of the objectives is to explore and propose a modular fusion architecture/framework for IMS based on cameras and laser sensors. In table 2.5 a summary of aforementioned developments is presented, including proposed work.

[1] http://www.poss.pku.edu.cn/index.html
[2] http://ko-fas.de/english/ko-per—cooperative-perception.html

| References | Project | Sensors | Target | Fusion | Framework |
|---|---|---|---|---|---|
| [70] | POSSi | Laser | Vehicles, pedestrians | No | No |
| [58] | POSSi | Lasers | Pedestrians | Yes | No |
| [68] | POSSi | Lasers | Vehicles, pedestrians | Yes | No |
| [67] | POSSi | Lasers, Camera | Vehicles, pedestrians | Yes | No |
| [69] | POSSi | Lasers | Vehicles, pedestrians | Yes | No |
| [29] | Ko-PER | Lasers, Cameras | Vehicles, pedestrians | Yes | No |
| [48] | Ko-PER | Lasers | Pedestrians | Yes | No |
| [49] | Ko-PER | Lasers | Vehicles, pedestrians | Yes | No |
| [52, 50] | Ko-PER | Lasers, Cameras | Vehicles, pedestrians | Yes | No |
| [59] | Ko-PER | Cameras | Vehicles | Yes | No |
| [51] | Ko-PER | Lasers, Cameras | Vehicles, pedestrians | Yes | No |
| **Proposed work** | **-** | **Lasers, Cameras** | **Vehicles** | **Yes** | **Yes** |

Table 2.5: Summary of recent developments involving sensor fusion for Intersection Management Systems

## 2.4 Conclusions

Multisensor data fusion can be defined as the process of combine, merge or integrate data from homogeneous or heterogeneous set of sensors in order to get a better representation of a process, the environment or a situation, through the inference of underlying information and the improve of quality of data. Depending of the nature of the sensors and the source of information, fusion can be done in several manners, i.e., sensor-level fusion, feature-level fusion or decision-level fusion.

Several models, architectures and frameworks have been proposed in literature; some of them to show how a data fusion system should work and some other providing guidelines on how to implement it in a given application. Also, a wide range of algorithms have been used to perform fusion of data, some classified by the nature of the fusion or by the nature of the data, but is finally the environment and the application itself which determines the approach to use.

On the other hand, intersection management systems are highly required to improve safety and mobility in transportation, due to the high complexity present in these places for drivers and for authorities too. There is not an

IMS application that address all the needs and problems in an intersection, for this reason many developments with narrow scopes and with many topics involved have been proposed through the years. To handle all these works and proposals, a new component-based classification scheme for IMS application has been proposed, and a review of developments in Intersection Management Systems is presented.

# Chapter 3

# Architecture Description and System Specification

”Perfect is enemy of good”

Voltaire

Although several types of sensors are used for intersection monitoring and supervision, the use of cameras, lasers and lidars has increased due to advances in sensors manufacturing and computing capabilities. Such enhancements allows to deploy more of those types of sensors per scenario and it is required to define some processing stages from raw data capture through decision and control stages. It is also needed to test and validate the developments prior to a real and full functional implementation. The first part of this chapter describes the main stages in a intersection management system, from data preprocessing to situation assesment. Then, two validation tools for IMS applications are described, simulation models and datasets. And finally, the proposed architecture for the implementation of an IMS system is presented.

## 3.1 Multisensor IMS

Multicamera and multilasers monitoring systems offer more information about environment that can be merged to provide a better representation of the whole scene, detect with more accuracy the objects in the intersection, and prevent possible incidents. For designing a single-sensor or multi-sensor IMS, there are some basic processing stages to have into account. In the case of a multisensor system, it is also required to analyse and determine which is the better fusion approach to use and in which of the processing stages this fusion should be performed, in order to get better results than a single-sensor based system.

### 3.1.1 Processing Stages

In the designing of an IMS, there are four main stages that have to be performed from the data source to final output: preprocessing, feature analysis, pattern recognition and situation assessment. The aim of the first stage is to extract data of interest from the raw sensor information, using filtering to remove noise and irrelevant data, and background subtraction techniques to get the foreground of the scene. Spatial-temporal alignment of data is also performed in this stage. In the second stage, the objective is to identify elements within the foreground and extract relevant features of them. The third stage receives the set of features from the previous stage and performs recognition and classification tasks. Also, tracking and prediction of objects' state is performed based on historic information. In the fourth stage, object behaviour and inter-objects interaction are analysed to identify context and detect situations or events of interest. This output could be delivered to an optional fifth stage of decision and control, to a human operator, or to a traffic agent or institution, to take immediate actions on traffic control, issue traffic tickets, warn drivers about possible incidents or improve transportation policies in a long-term basis. In figure 3.1, previously described stages are depicted, including a list of common tasks performed at each of these stages and also it is shown how the data volume is reduced while data meaning increases in the last stages.

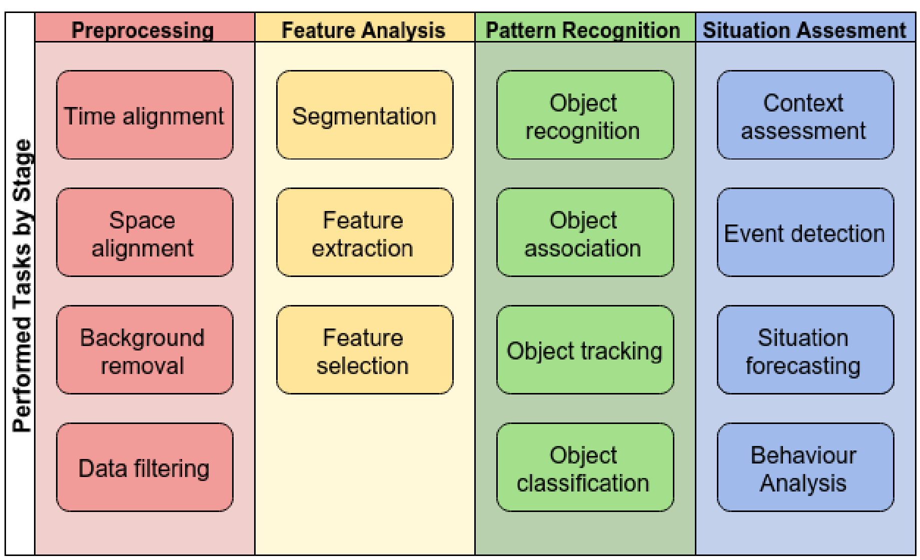


Figure 3.1: Processing stages in an IMS and commonly performed tasks within each of them

### 3.1.2 Fusion approaches

Depending on wheter the system has multiple sensors of the same type or different type, a data fusion approach should be chosen. When data came from same-type sensors, it is usually fused at low levels using techniques for temporal-spatial alignment, depending on sensors configuration. This is the case for a network of lasers or lidars or for multi-camera systems. If the system have different type of sensors, data from them may be fused at mid level, based on extracted features or classes on each subsystem; or may be fused at high level if each subsystem delivers control or decision outputs. The tasks shown in each of the processing stages (Figure 3.1), could be used as fusion blocks for homogeneous or heterogeneous data.

## 3.2 Validation approaches

Before deploying an IMS, it is needed to test some of its components and validate their results. Two commonly used tools for this purpose are traffic simulation models and datasets, depending on the component to be tested. Below, a description of each one is presented.

### 3.2.1 Transportation and Traffic Simulation

Transportation is a highly complex activity where different elements, like infrastructure, vehicles and pedestrians, affect efficiency, safety and quality of traffic. Intersections are special cases because there exists a high interaction between those mentioned elements making of these places critical points for mobility. For this reason, it is needed to have traffic models to allow the simulation of new policies or deployments intended to enhance transportation.

Three widely-known models for this kind of analysis are macroscopic model, microscopic model and mesoscopic model. The macroscopic model of traffic flow is based on a hydrodynamic analogy, modeling traffic as a fluid process characterized by three main variables, density, volume and speed, and the objective is to describe time-space evolution of those variables.

Simulators description (?)

A more detailed analysis of traffic simulation models and simulation platforms could be found in [2, 11, 38, 41]

### 3.2.2 Intersection monitoring Datasets

As mentioned in section 2.3, POSS-i and Ko-PER projects are leading the development of multisensor Intersection Management Systems. One of the contributions of these projects is the creation of datasets of such systems. They provide camera and laser information of a monitored intersection in Peking, China and Aschaffenburg, Germany, respectively. Next, a description of these two datasets will be given.

#### POSSi Dataset

POSSi dataset[1]

Dataset description (?)

#### KoPER Dataset

Ko-PER dataset[2]

The full description of this dataset is presented in [60].

## 3.3 Architecture proposal

The proposal presented in this work is based on MFI model processing entities in the sense that a processing block could take one or more inputs related between them and generate an output of the same type or a higher level output. The communication or data exchange approach is based on JDL model, in which data is available over a "bus", where the processing blocks can write to or read from. Four main components of the architecture are defined: communication scheme, data model, information structure and processing blocks.

### 3.3.1 Communication scheme

In order to isolate the processing tasks of the communication system, a Publisher-Subscriber approach is selected. The benefits of this option is that data can be shared between diferent blocks without generating a dependency from publishers to subscribers. Another benefit is that exposing a defined mechanism for publishing or subscribing, allows the system to be implementation agnostic.

Taking these reasons into account, Redis is chosen as communication platform. As stated in their website, "Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker"[57] which support Publisher-Subscriber paradigm, allowing greater scalability and a more dynamic network topology.

For publishers and subscribers to exchange messages, channels are defined with labels using a namespace approach that describe the source of data and the kind of data. For example a channel containing raw data from a laser scanner could be labeled as `/sensors/range/laser_1/data/raw`.

Redis is also used as in-memory storage for configuration data that is not part of the processing flow, like sensor parameters, intersection model and control options. This information is loaded into Redis from a JSON file and it is available for any client requiring it.

[1] Available at http://www.poss.pku.edu.cn/download.html
[2] Available at http://www.uni-ulm.de/in/mrm/forschung/datensaetze.html

### 3.3.2 Data Model

The data model proposed for the system includes the abstraction of the sensing elements and the intersection model. For the sensing elements, a Sensor base class and two derived classes, RangeSensor class and ImageSensor class, are defined.

The intersection has been modeled as a class with attributes like a label, a Map object, a set of RangeSensors and CameraSensors objects, a set of Leg objects, and a set of Area objects, for inner areas of the intersection. The Map class contains information about the geometry of the intersection and the coordinate system, including a image intended for visualization of configuration and processing.

The sensors sets are composed of zero or more sensor objects, as defined previously. An Area class is defined as having a bounding box and a label . From this class, a derived class, Leg, is defined for containing additional information about the legs of the intersection, such as heading of the leg, approaching or departure type. In figure 3.2 is shown an UML diagram for the aforementioned classes and their relationships

The purpose of all the labels attributes within each class is to serve as an identifier of the channel in which the entity is publishing or subscribing to; in other words, to identify where is data generated from or where is data going to. For example, channel `/sensors/range/laser_1/data/raw` contains raw data from a laser scanner labeled as `"laser_1"` and `/legs/leg_3/occupancy/` refers to the occupancy level of the leg labeled as `leg_3`.

### 3.3.3 Information Structure

Data exchanged between processing blocks or any paramater stored in Redis, is formatted using JavaScript Object Notation (JSON), allowing to use any JSON parser/formatter available in many languages, even it is possible to create simple scripts to interact with the system.

#### .imscfg file

For running, the systems requires first a .imscfg file which describes the scenario and the sensors configuration. This information is stored as a JSON object with the properties decribed in table 3.1.

#### data messages and configuration parameters

In addition to .imscfg file, there are different types of messages and parameters which have its own structure, for example used for sensors configuration or processed data. The table 3.2 gives a description of the types of information used. A more detailed documentation is given in appendix TODO.

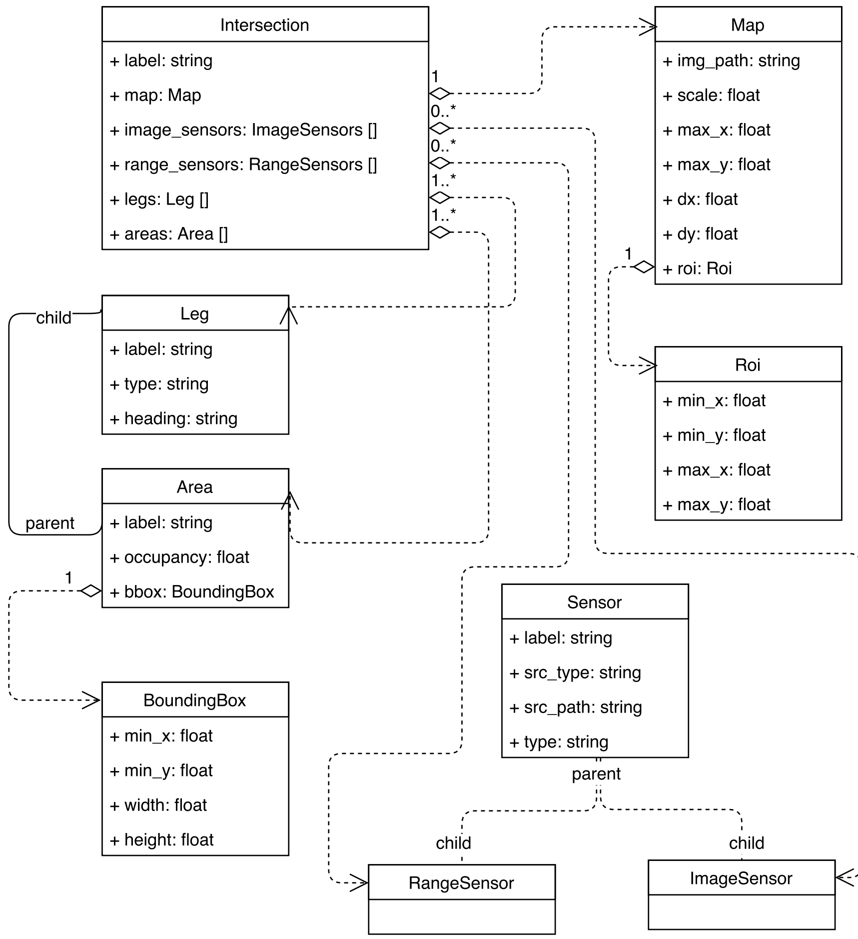


Figure 3.2: UML diagram of defined entities and their relationships

| Property | Description |
|---|---|
| name | Name of the configuration |
| map | Information of the scene, including map, coordinate system and region of interest. |
| cameras | List of camera sensors information |
| range_sensors | List of range sensors information |
| legs | List of legs information |
| intersection | Information of area of the intersection |

Table 3.1: Properties in .imscfg JSON file

| Name | Description |
|---|---|
| laser_pol_msg | Contains a timestamp, an array of N angles and and array of N measurements |
| laser_cart_msg | Contains a timestamp, an array of N (x, y) coordinates |
| laser_cfg | Contains position and orientation information, also has a background model for the laser sensor |
| occgrid_msg | Contains a timestamp and an occupancy grid in the form of an MxN array |
| occgrid_cfg | Contains occupancy grid configuration for the scene |
| clustering_msg | Contains a set of clusters, each of them with an ID and a set of points |
| clustering_cfg | Contains configuration about clustering algorithm and parameters |

Table 3.2: Description of diferent types of messages and parameters used

### 3.3.4 Processing Blocks

As stated in previous section, there are defined four stages in the processing flow, namely, preprocessing, feature analysis, pattern recognition and situation assessment. Within each of these stages there are different methods and techniques used to process the data and it is possible that those methods are suitable to perform fusion between homogeneous or heterogeneous data.

Below are detailed different processing blocks implemented as part of the whole architecture proposal, classified by the stage of processing in which are located in the process flow. Also, it is shown the format of data input, data output and parameters needed for configuration and execution.

Additionaly, there are included some tools and generic blocks used for dataset reading, data visualization and interactive control.

#### Preprocessing

**laser_background_remove**

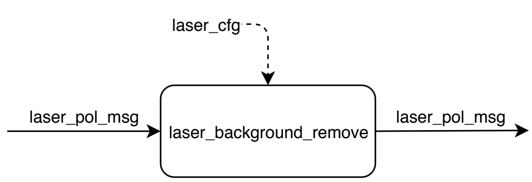


Figure 3.3: laser_bg_remove

This block takes as input a message of type laser_pol_msg, which contains raw readings from laser sensor. Also, it takes a laser_cfg parameter which includes a background model for the laser sensor, generated from a set of scans and taking the maximum measurement in each angle of scanning, defined as follows:

Let suppose we have N scans from the laser and $d_{\theta i}$ be the distance measure at angle $\theta$ in scan $i$. Thus, the background value for that angle $\theta$ is $bg_\theta = max(\{d_{\theta i} | 1 < i < N\})$ and the background model is $bg = \{bg_\theta | \theta \in \Theta\}$ where $\Theta$ is the set of angles of scanning, in this case from -90°to 90°with 0.5°step.

### laser_polar2cart

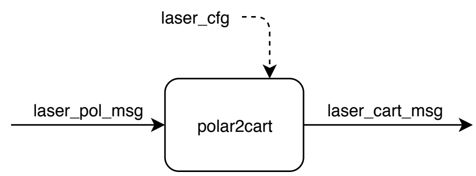


Figure 3.4: polar_to_cartesian

This block takes as input a message of type laser_pol_msg for converting it into cartesian coordinates referenced to global system. For this reason, it also takes a laser_cfg parameter to include laser position information in the conversion. Output message contains a set of x, y points obtained from the following equations:

$$\begin{aligned} O_x &= I_\rho * \cos\phi + S_x \\ O_y &= I_\rho * \sin\phi + S_y \\ \phi &= I_\theta + S_\theta \end{aligned}$$

Where:

$(S_x, S_y, S_\theta)$: Position and orientation of the laser
$(I_\rho, I_\theta)$: Input message (distance and angle arrays)
$(O_x, O_y)$: Output message (x and y arrays)

### laser_cart_merge

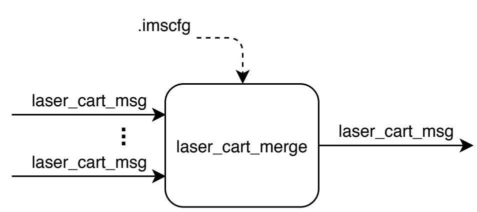


Figure 3.5: laser_cart_merge

This block takes as input messages of type laser_cart_msg coming from different sensors and merges them into a single message, based on a sampling period $T$.

### Feature Analysis

**points_to_cluster**

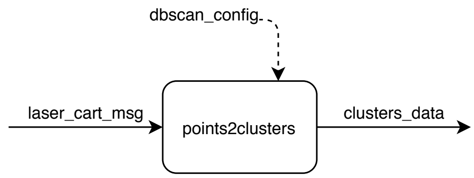


Figure 3.6: points_to_clusters

The input message for this block is a set of points corresponding to objects scanned by lasers. Clustering is performed over this input to identify the points belonging to the same object. The algorithm used in this implementation is DBSCAN, which stands for Density-Based Spatial Clustering of Applications with Noise. This algorithm does not need an estimated number of clusters as input, instead of this, it requires only two parameters: a minimum number of points per cluster, m, and a neighbourhood measure, $\epsilon$. A detailed description of the algorithm, can be found in [27].

As output, a message of type clustering_msg, containing information like clusters ID and set of points belonging to each cluster, is delivered.

**points_to_occgrid**

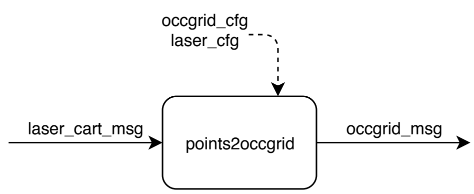


Figure 3.7: points_to_grid

This block generates a occupancy grid based on cartesian data from laser. As parameters, this block receives a cell size for the grid, the laser configuration and the scene configuration. The grid size is of $N_C$ Columns and $N_R$ rows, obtained as follows:

$$\begin{aligned}
X &= x_{max} - x_{min} \\
Y &= y_{max} - y_{min} \\
N_C &= ceil(X/C) + 1 \\
N_R &= ceil(Y/C) + 1
\end{aligned}$$

Where $(x_{min}, y_{min})$ and $(x_{max}, y_{max})$ are the bottom-left and top-right coordinates of the region of interest in the scene, respectively.

Each cell of the grid will store a value indicating a probability of that cell being occupied. Initially, all cells have a value of 0.5. In order, to update the grid using data from laser, the points corresponding to laser position and measures, should be located in a cell of the grid, and then mark it as occupied. The following equations show how a point $P = (P_x, P_y)$ is tranformed to a cell $C_{ij}$, where $i$ is the column and $j$ is the row in the grid:

$$
\begin{aligned}
i &= ceil((P_x - x_{min})/C - 0.5) \\
j &= N_R - ceil((P_y - y_{min})/C - 0.5)
\end{aligned}
$$

Then, the cells that form a straigh line between the cell of laser position and the cells of each laser measure, should be marked as empty. To get the list of these cells Bresenham's algorithm is used. This algorithm is widely used in computer graphics due to it simplicity and because it uses integer arithmetics, making it computationally cheap.

Check bresenham algorithm reference

The input of this algorithm is a pair of coordinates, the endpoints of the line, and as result, the set of points that completes the line between, is returned.

Now, the update procedure for the probability in each cell is described for the function $f$:

update policy

**Pattern Recognition**

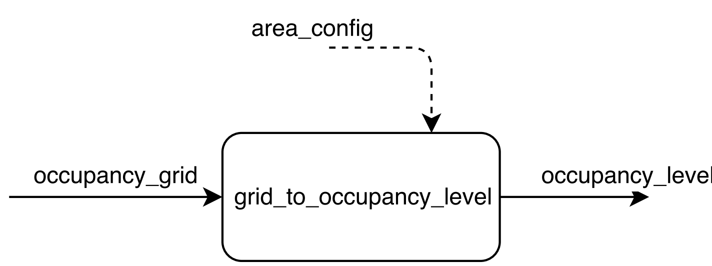


Figure 3.8: grid_to_occupancy_level

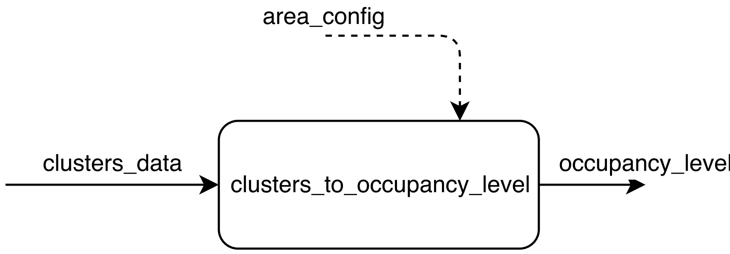


Figure 3.9: clusters_to_occupancy_level

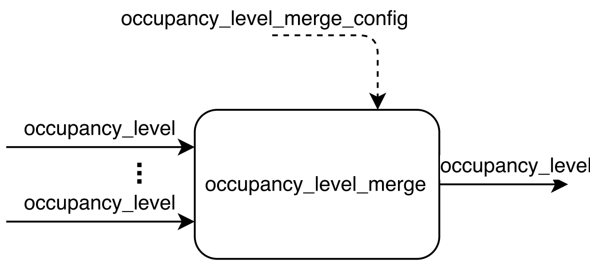


Figure 3.10: occupancy_level_merge

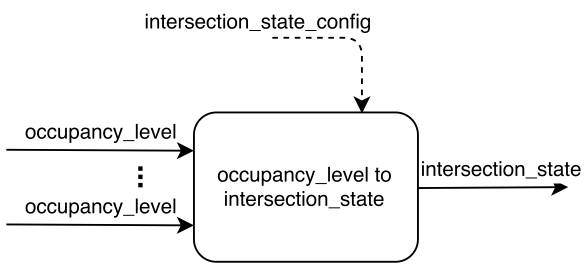


Figure 3.11: occupancy_level_to_intersection_state

#### Situation Assesment

### Tools and Utilities

## 3.3.5 Conclusions

# Chapter 4

# Deployment and Results

## 4.1 Test Configurations

In order to test the proposed architecture, three different configurations have been selected. The first configuration is using just one laser sensor. The second configuration is based on multiple laser sensors, Finally, the third configuration uses multiple lasers along with a camera.

Each configuration consists of a set of processing blocks (as described in section 3.3.4) connected, aiming to take data from sensors, lasers and cameras, to produce an output of higher level. In this case, vehicle counting and intersection occupation level are the metrics used for evaluation.

The following table lists all used blocks and assigns an ID for each one. Those IDs are used in the graph description in its own section. Also, bold nodes and connections indicate multiple instance of the same element.

### 4.1.1 Case 1: Single Laser

TODO

### 4.1.2 Case 2: Multiple Lasers

TODO

### 4.1.3 Case 3: Multiple Lasers and camera

TODO

### 4.1.4 Results and Comparison

TODO

| Block ID | Name |
|---|---|
| A | laser_publisher |
| B | laser_bg_remover |
| C | laser_pol2cart |
| D | laser_cart_merge |
| E | points2clusters |
| F | clusters_leg_counter |
| G | leg_counter_merge |
| H | clusters2occ_level |
| I | occ_level_merge |
| J | laser_pol2cart |
| K | points2occgrid |
| L | occgrid_merge |
| M | occgrid_leg_counter |
| N | occgrid2occ_level |
| O | camera_publisher |
| P | camera_bg_remover |
| Q | camera_object_filter |
| R | camera_object2occgrid |

Table 4.1: Description of processing blocks used in test configurations

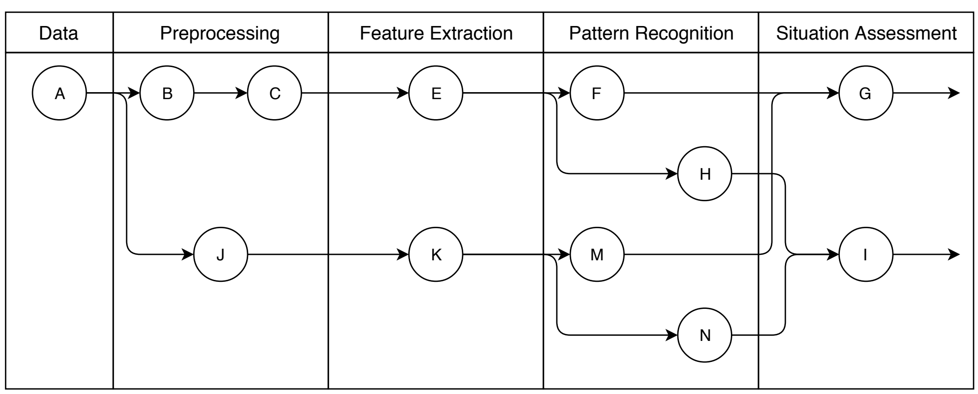


Figure 4.1: Single laser configuration

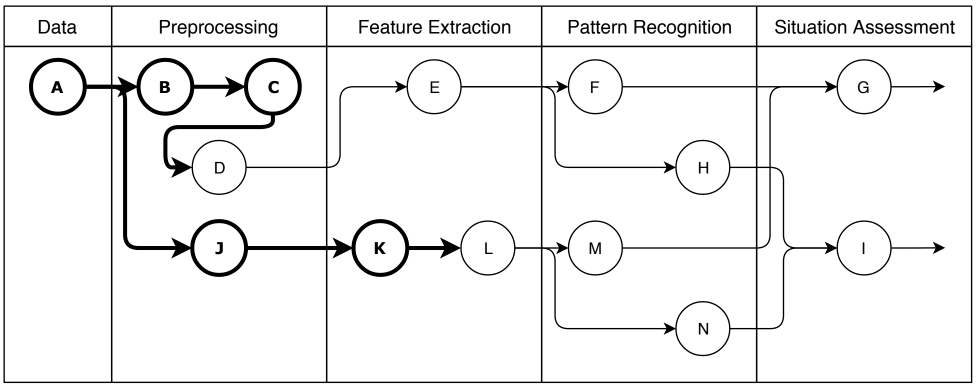


Figure 4.2: Single laser configuration

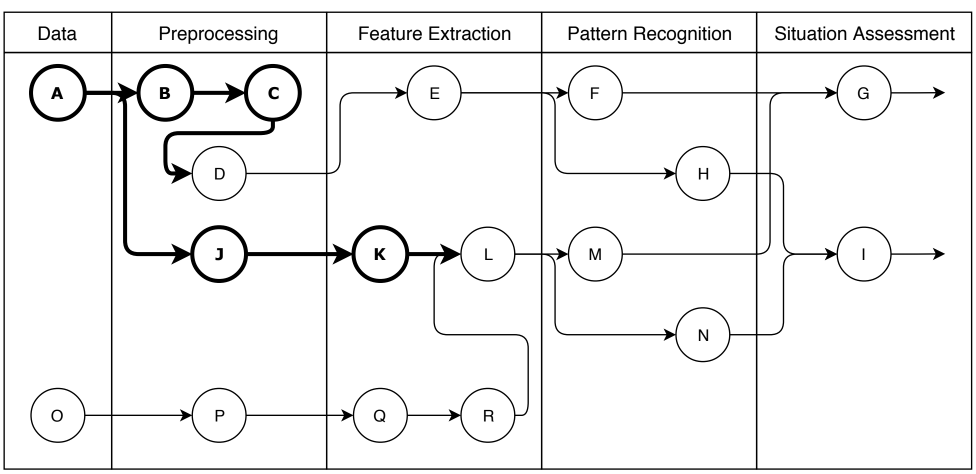


Figure 4.3: Single laser configuration

# Chapter 5

# Conclusions

## 5.1 General Conclusions

## 5.2 Contributions

A scalable, modular system for traffic control monitoring at intersection. Both for experimentation and real deployment.

Set of software repositories for application and for available datasets containing intersection data.

Pair of articles submmited to events, one for architecture description and example implementation and another for application comparison.

## 5.3 Future Work

Generation of our own dataset.

Optimization of communication scheme.

Integration of new types of sensors

Creation of a more friendly user interface.

# Appendices

# Appendix A

# Vehicular Environment Simulators

# Bibliography


[1] M. Abbas and P. Li. Advanced real-time intersection data collection system for safety evaluation. *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pages 1–6, Oct. 2009.

[2] S. Adams Boxill and L. Yu. An Evaluation of Traffic Simulation Models for Supporting ITS Development. Technical report, Texas Southern University, Houston, 2000.

[3] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An Architecture for Autonomy, 1998.

[4] S. Atev, H. Arumugam, O. Masoud, R. Janardan, and N. Papanikolopoulos. A Vision-Based Approach to Collision Prediction at Traffic Intersections. *IEEE Transactions on Intelligent Transportation Systems*, 6(4):416–423, Dec. 2005.

[5] T.-c. Au, S. Korea, S. Zhang, and P. Stone. Autonomous Intersection Management for Semi-Autonomous Vehicles. 2015.

[6] A. Avila, G. Korkmaz, Y. Liu, H. Teh, E. Ekici, F. Ozguner, U. Ozguner, K. Redmill, O. Takeshita, K. Tokuda, M. Hamaguchi, S. Nakabayashi, and H. Tsutsui. A complete simulator architecture for inter-vehicle communication intersection warning systems. In *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pages 461–466, Vienna, Austria, 2005. Ieee.

[7] R. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige. Intersection Management using Vehicular Networks. pages 1–13, Apr. 2012.

[8] P. Babaei. Vehicles tracking and classification using Traffic zones in a hybrid scheme for intersection traffic management by smart cameras. In *2010 International Conference on Signal and Image Processing ICSIP*, pages 49–53. Islamic Azad University, West Tehran Branch, Iran, Ieee, 2010.

[9] P. Babaei and M. Fathy. Abnormality Detection and Traffic Flow Measurement Using a Hybrid Scheme of SIFT in Distributed Multi Camera Intersection Monitoring. *2011 7th Iranian Conference on Machine Vision and Image Processing*, pages 1–5, Nov. 2011.

[10] R. Ball and N. Dulay. Enhancing Traffic Intersection Control with Intelligent Objects. *Urban IOT 2010 Urban Internet of Things Towards Programmable Realtime Cities*, 2010.

[11] J. Barcelo, editor. *Fundamentals of Traffic Simulation*. Springer, 2000.

[12] F. Basma, Y. Tachwali, and H. H. Refai. Intersection collision avoidance system using infrastructure communication. *2011 14th International IEEE Conference on Intelligent Transportation Systems ITSC*, pages 422–427, 2011.

[13] M. Bedworth and J. O'Brien. The Omnibus model: A new model of data fusion? *IEEE Aerospace and Electronic Systems Magazine*, 15:30–36, 2000.

[14] H. Boström, S. Andler, and M. Brohede. On the definition of information fusion as a field of research. Technical report, University of Skövde, 2007.

[15] J. R. Boyd. A discourse on winning and losing. 1987.

[16] C.-Y. Chan and B. Bougler. Evaluation of cooperative roadside and vehicle-based data collection for assessing intersection conflicts. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 165–170. IEEE, 2005.

[17] C.-y. Chan, D. Marco, and R. F. Station. Traffic Monitoring at Signal-Controlled Intersections and Data Mining for Safety Applications. pages 355–360, 2004.

[18] E.-H. Choi. Crash Factors in Intersection-Related Crashes: An On-Scene Perspective. Technical Report September, U.S. Departament of Transportation, Springfield, VA, 2010.

[19] L. Conde Bento, R. Parafita, and U. Nunes. Intelligent traffic management at intersections supported by V2V and V2I communications. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1495–1502, Anchorage, Alaska, USA, 2012.

[20] Corporación Fondo de Prevención Vial. Anuario Estadístico de Accidentalidad Vial. Technical report, Bogotá, Colombia, 2010.

[21] A. Dogan, G. Korkmaz, Y. Liu, F. Ozguner, U. Ozguner, K. Redmill, O. Takeshita, and K. Tokuda. Evaluation of intersection collision warning system using an inter-vehicle communication simulator. In *2004 IEEE Intelligent Transportation Systems Conference*, pages 1103–1108, Washington, D.C., USA, 2004. Ieee.

[22] K. Dresner and P. Stone. A Multiagent Approach to Autonomous Intersection Management. *Journal of Artificial Intelligence Research*, 31:591–656, 2008.

[23] J. Eguchi and H. Koike. Discrimination of an Approaching Vehicle at an Intersection Using a Monocular Camera. In *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, 2007.

[24] W. Elmenreich. A Review on System Architectures for Sensor Fusion Applications. In *Software Technologies for Embedded and Ubiquitous Systems Lecture Notes in Computer Science Volume 4761, 2007,*, pages 547–559, 2007.

[25] W. Elmenreich and S. Pitzek. Using sensor fusion in a time-triggered network. *IECON'01. 27th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.37243)*, 1, 2001.

[26] J. Esteban, A. Starr, R. Willetts, P. Hannah, P. Bryanston-Cross, and N. Comput. A Review of data fusion models and architectures: towards engineering guidelines. *Neural Computing and Applications*, 14(4):273–281, June 2005.

[27] M. Ester, H.-p. Kriegel, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.

[28] M. Friesen, R. Jacob, P. Grestoni, T. Mailey, and R. McLeod. VEHICULAR TRAFFIC MONITORING USING BLUETOOTH. In *2013 26th IEEE Canadian Conference Of Electrical And Computer Engineering (CCECE)*, 2013.

[29] M. Goldhammer, E. Strigel, D. Meissner, U. Brunsmann, K. Doll, and K. Dietmayer. Cooperative multi sensor network for traffic safety applications at intersections. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1178–1183. IEEE, Sept. 2012.

[30] J. A. Guerrero-Ibañez, C. Flores-Cortés, J. M. Ramírez-Alcaraz, P. Santana, T. Mendoza-Robles, H. A. Vizcaíno-Anaya, E. Peña Cárdenas, and A. Anguiano-Mancilla. Sistema Inteligente basado en comunicación V2X para prevención de colisiones en intersecciones viales. In *Décima Segunda Conferencia Iberoamericana en Sistemas, Cibernética e Informática. CISCI 2013*, Orlando, FL, 2013.

[31] D. L. Hall and J. Llinas. An Introduction to Multisensor Data Fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[32] C. J. Harris, A. Bailey, and T. J. Dodd. Multi-sensor data fusion in defence and aerospace. *Aeronautical Journal*, 102:229–244, 1998.

[33] B. Jarupan, Y. Balcioglu, E. Ekici, F. Özguner, and U. Özguner. An Integrated Wireless Intersection Simulator for Collision Warning Systems in Vehicular Networks. In *Proceedings of the 2008 IEEE International Conference on Vehicular Electronics and Safety*, pages 340–345, Colombus, USA, 2008.

[34] Q. Jin, G. Wu, K. Boriboonsomsin, and M. Barth. Advanced Intersection Management for Connected Vehicles Using a Multi-Agent Systems Approach. In *2012 Intelligent Vehicles Symposium*, Alcalá de Henares, Spain, 2012.

[35] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic Monitoring and Accident Detection at Intersections. In *IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, 1999.

[36] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):108–118, June 2000.

[37] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, Jan. 2013.

[38] R. Kitamura and M. Kuwahara, editors. *Simulation Approaches in Transportation Analysis*. Springer, 2005.

[39] R. Kumar, M. Wolenetz, B. Agarwalla, J. Shin, P. Hutto, A. Paul, and U. Ramachandran. DFuse: a framework for distributed data fusion. *Conference On Embedded Networked Sensor Systems*, 2003.

[40] K. KURAMOTO, K. FUJIMURA, and T. HASEGAWA. The Multi-Class Zone ITS Communication Scheme for Real-time Communications in Intersections. In *Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference*, pages 431–435, 2007.

[41] E. Lieberman and A. K. Rathi. Traffic Simulation. In N. H. Gartner, C. J. Messer, and A. K. Rathi, editors, *Traffic Flow Theory*, chapter 10. FHWA, 2 edition, 1992.

[42] J. Llinas and D. L. Hall. An introduction to multi-sensor data fusion. *ISCAS '98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (Cat. No.98CH36187)*, 6:537–540, 1998.

[43] R. Luo. Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE Sensors Journal*, 2(2):107–119, Apr. 2002.

[44] R. Luo and M. Kay. A tutorial on multisensor integration and fusion. In *[Proceedings] IECON '90: 16th Annual Conference of IEEE Industrial Electronics Society*, pages 707–722. IEEE, 1990.

[45] R. C. Luo, C. C. Chang, and C. C. Lai. Multisensor Fusion and Integration: Theories, Applications, and its Perspectives. *IEEE Sensors Journal*, 11(12):3122–3138, Dec. 2011.

[46] R. C. Luo and M. G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man and Cybernetics*, 19:901–931, 1989.

[47] D. Meissner and K. Dietmayer. Simulation and calibration of infrastructure based laser scanner networks at intersections. *2010 IEEE Intelligent Vehicles Symposium*, pages 670–675, June 2010.

[48] D. Meissner, S. Reuter, and K. Dietmayer. Real-time detection and tracking of pedestrians at intersections using a network of laserscanners. *2012 IEEE Intelligent Vehicles Symposium*, pages 630–635, June 2012.

[49] D. Meissner, S. Reuter, and K. Dietmayer. Combining the 2D and 3D world: a new approach for point cloud based object detection. In *IET Intelligent Signal Processing Conference 2013 (ISP 2013)*, pages 4.1–4.1. Institution of Engineering and Technology, 2013.

[50] D. Meissner, S. Reuter, and K. Dietmayer. Road user tracking at intersections using a multiple-model PHD filter. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, number Iv, pages 377–382. IEEE, June 2013.

[51] D. Meissner, S. Reuter, E. Strigel, and K. Dietmayer. Intersection-Based Road User Tracking Using a Classifying Multiple-Model PHD Filter. *IEEE Intelligent Transportation Systems Magazine*, 6(April 2014):21–33, Jan. 2014.

[52] D. Meissner, S. Reuter, B. Wilking, and K. Dietmayer. Road user tracking using a Dempster-Shafer based classifying multiple-model PHD filter. In *Information Fusion (FUSION), 2013 16th International Conference on*, volume 32, pages 1236–1242, Nov. 2013.

[53] S. Messelodi, C. M. Modena, and M. Zanin. A computer vision system for the detection and classification of vehicles at urban road intersections. Technical report, 2004.

[54] R. S. Mundewadikar, S. S. Dorle, A. G. Keskar, and M. B. Chakole. Intersection Collision Detection and Warning Protocol: Design Approach. In *First International Conference on Emerging Trends in Engineering and Technology*, pages 869–872. Ieee, 2008.

[55] M. Quinlan, T.-C. A. T.-C. Au, J. Zhu, N. Stiurca, and P. Stone. Bringing simulation to life: A mixed reality autonomous intersection. *Intelligent Robots and Systems IROS 2010 IEEERSJ International Conference on*, (October):6083–6088, 2010.

[56] Z. Y. Rawashdeh and S. M. Mahmud. Intersection Collision Avoidance System Architecture. *2008 5th IEEE Consumer Communications and Networking Conference*, pages 493–494, 2008.

[57] redislabs. Introduction to redis. https://redis.io/topics/introduction. Accessed: 2018-01-22.

[58] X. Song, J. Cui, H. Zhao, and H. Zha. Bayesian fusion of laser and vision for multiple People Detection and tracking. *2008 SICE Annual Conference*, pages 3014–3019, Aug. 2008.

[59] E. Strigel, D. Meissner, and K. Dietmayer. Vehicle detection and tracking at intersections by fusing multiple camera views. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 882–887. IEEE, June 2013.

[60] E. Strigel, D. Meissner, F. Seeliger, B. Wilking, and K. Dietmayer. The Ko-PER Intersection Laserscanner and Video Dataset. In *IEEE 17th International Conference on Intelligent Transportation Systems*, pages 1900–1901, Qingdao, 2014.

[61] J. M. Sussman. *Perspectives on Intelligent Transportation Systems (ITS).* Springer US, Boston, MA, 2005.

[62] H. Veeraraghavan, O. Masoud, and N. Papanikolopoulos. Vision-based Monitoring of Intersections. (September):7–12, 2002.

[63] H. Veeraraghavan, O. Masoud, and N. P. Papanikolopoulos. Computer vision algorithms for intersection monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 4(2):78–89, 2003.

[64] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos. Switching Kalman Filter-Based Approach for Tracking and Event Detection at Traffic Intersections. *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pages 1167–1172, 2005.

[65] F. E. White. Data Fusion Lexicon. In *The Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3*, volume 15, page 15, 1991.

[66] B. Williams. *Intelligent Transport Systems Standards.* Artech House, INC., 2008.

[67] H. Zhao, J. Cui, and H. Zha. Sensing an Intersection Using a Network of Laser Scanners and Video Cameras. *IEEE Intelligent Transportation Systems Magazine*, pages 31–37, 2009.

[68] H. Zhao, J. Cui, H. Zha, K. Katabira, X. Shao, and R. Shibasaki. Monitoring an intersection using a network of laser scanners. In *Proceedings of the11th International IEEE Conference on Intelligent Transportation Systems*, pages 428–433, Beijing, China, 2008. Ieee.

[69] H. Zhao, J. Sha, Y. Zhao, J. Xi, J. Cui, H. Zha, and R. Shibasaki. Detection and Tracking of Moving Objects at Intersections Using a Network of Laser Scanners. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):1–16, 2012.

[70] H. Zhao and R. Shibasaki. Joint tracking and classification of moving objects at intersection using a single-row laser range scanner. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 287–294, Toronto, Canada, 2006.