Contributions of context-aided multi-modal perception systems for detection and tracking of moving objects

Egor Sattarov

NNT: 2016SACLS354

Thèse de Doctorat de l'Université Paris-Saclay préparée à l'Université Paris-Sud XI

Ecole Doctorale №580

Sciences et technologies de l'information et de la communication

Spécialité de doctorat: Traitement du Signal et des images

par

M. Egor Sattarov

Etude et quantification de la contribution des systèmes de perception multimodale assistés par des informations de contexte pour la détection et le suivi d'objets dynamiques

Thèse présentée et soutenue à Gif-sur-Yvette, le 9 décembre 2016

Composition du Jury:

M. Filliat David	Professeur, ENSTA ParisTech	Président
M. Martinet Philippe	Professeur, IRCCYN	Rapporteur
M. Kummert Franz	Professeur, Universität Bielefeld	Rapporteur
Mme. Cherfaoui Véronique	Maître de conférences, UTC	Examinatrice
M. Ibanez Javier	Directeur de recherche, Renault R.	Examinateur
M. Reynaud Roger	Professeur, Université Paris Sud	Directeur de thèse
M. Rodríguez Sergio	Professeur, Université Paris Sud	Co-encadrant de thèse
M. Gepperth Alexander	Professeur, ENSTA ParisTech	Co-encadrant de thèse

Contents

Notations

Common notations

X, x	Scalar values or sets
$\mathbf{X}$	Matrix
$\mathcal{X}$	Space
$\mathbf{x}$	Vector

Multi-sensor association

$\mathbf{p}$	Prototype vector associated to a SOM node
$\mathcal{N}()$	Gaussian distribution
ω	Weight associated between nodes of two SOMs
h	Neuron activity on SOM prototype given by detection $\mathbf{z}$

Spaces

$\mathcal{ECEF}$	ECEF space
$\mathcal{ENU}$	ENU space
$\mathcal{LLA}$	LLA space
$\mathcal{L}$	Space of LIDAR detections
$\mathcal{P}$	State space for tracking
$\mathcal{T}$	State space for tracking where context is defined
$\mathcal{V}$	Space of vision detections
$\hat{\mathcal{X}}$	Complement space, for association
$\overline{\mathcal{L}}$	Space of pseudo-LIDAR for GPS-LIDAR transform

Tracking

$G()$	Pseudo-distance function
R_d, R_b, R_{ret}, R_m	Tracker parameters: death/birth/retard/life rates
XZ^{assoc}	Set of pairs "track-detection" for association
X	Set of tracks
Z	Set of detection

$\mathbf{x}$	Track state
$\mathbf{y}$	Ground Truth state
$\mathbf{z}$	Detection
θ	Thresholds
ξ	Particle
c, d, v	Track's characteristics: position, size and speed
k	Discrete time index
w	Particle's weight

Transforms

$\mathbf{A}$	Camera intrinsic parameters matrix
$\mathbf{R}$	Rotation matrix
$\mathbf{T}$	Linear transform
$\mathbf{t}$	Translation vector
t	Continuous timestamp

Acronyms

ADAS	Advanced Driver Assistance Systems
CAN	Controller Area Network
CCD	Charge-Coupled Device
CMOS	Complementary Metal-Oxide Semiconductor
DATMO	Detection And Tracking of Moving Objects
DGPS	Differential Global Positioning System
ECEF	Earth-Centered, Earth-Fixed
ENU	East, North, Up
GBAS	Ground-Based Augmentation System
GIS	Geographic Information Sysem
GLONASS	GLObal NAvigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GT	Ground Truth
HOG	Histogram of Oriented Gradients
IGS	International GNSS Service
IHS	Intensity-Hue-Saturation
JPDA	Joint Probabilistic Data Association
JPDAF	Joint Probabilistic Data Association Filter
KLT	Kanade-Lucas-Tomasi
LIDAR	LIght Detection And Ranging
LLA	Latitude, Longitude, Altitude
MHKF	Multiple Hypothesis Kalman Filter
MHT	Multiple Hypothesis Tracking
MTT	Multiple Taret Tracking
NMEA	National Marine Electronics Association
OSM	Open Street Map

PCA	Principal Component Analysis
PF	Particle Filter
PHD	Probability Hypothesis Density
POSIX	Portable Operating System Interface
RADAR	RAdio Detection And Ranging
RFS	Random Finite Set
RGB	Red-Green-Blue
RGB-D	Red-Green-Blue-Depth
RINEX	Receiver INdependent EXchange
ROC	Receiver Operating Characteristic
ROI	Region Of Interest
ROS	Robot Operating System
RTK	Real Time Kinematic
SBAS	Satellite-Based Augmentation System
SIFT	Scale-Invariant Feature Transform
SMC	Sequential Monte-Carlo
SOM	Self-Organizing Map
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TCP	Transmission Control Protocol
ToA	Time of Arrival
ToF	Time-of-Flight
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
WGS-84	World Geodetic System
WSS	Wheel Speed Sensors

List of Figures

List of Tables

List of Algorithms

List of Listings

General introduction

The modern world is unthinkable without vehicles. Urban infrastructure in its current form is resulting from motor car necessity. As the number of vehicles increases, roads enlarge and extend. Human mobility is strongly coupled with personal cars - the major part of vehicles on the roads. Due to the cars domination in our life it is not surprising that the questions of autonomous driving and intelligent vehicles are rapidly evolving nowadays. The autonomous driving can reduce human-caused accidents and liberate the time of driving. Some of the key research problems of autonomous driving are self-positioning, environment understanding and traffic interactions with other vehicles and infrastructure.

Research domains

Scene understanding for an autonomous vehicle is a task which is closely related to functions like road marking detection, road signs recognition, obstacle detection and dynamic objects tracking. This last function is extremely important in road safety applications since it allows for collision avoidance, preventing or mitigating accident consequences. The object tracking itself is a large research subject. It has emerged motivated from military applications in the beginning of the 20th century. After large development of relatively cheap sensors in 70s-80s, the tracking methods were widely adopted and implemented on several applications. Multi-object tracking for intelligent vehicles is a complex problem since it deals with multiple environments (rural, semi-urban and urban), sensors limitations, multiple objects dynamics and few computational resources.

Tracking using one sensor is a non-trivial issue, multi-sensor tracking is an even more challenging problem since the information provided by several sensors must be fused. Data fusion is intended to enhance the precision and the confidence of the tracking estimates. That means that the combination of multiple sensors data can provide more specific inference than a single sensor. Modern autonomous cars projects always consider multiple sensors, like cameras, RADARs, LIDARs for environment perception; GPS, speed and inertial sensors for self-positioning. Multi-sensor multi-object tracking for autonomous urban driving is a dynamically developing area [H. Cho et al. 2014; Petrovskaya and Thrun 2009; Darms, Rybski, and Urmson 2008]. An example of a vehicle equipped with multiple sensors for tracking is shown in Fig. 1. Zoe is the experimental vehicle of the SATIE Laboratory, University Paris Saclay.

While the data fusion enhances tracking by taking advantage of the combination of raw observation data, a complementary approach consists in using additional, contextual (semantic or geometrical) information about the environment so as to improve the tracking process. The

contextual information can be extracted from the observed raw measurements, like road lane detection using vision or turns and stops at crossroads from Geographic Information Systems (GIS).

The thesis topic addressed in this manuscript is *context-aided multi-modal or multi-sensor systems for tracking dynamic objects.*

Figure 1 – ZOE - experimental platform of SATIE Laboratory, University Paris Saclay

Statement of the problem

Multi-sensor object tracking entails multiple challenges as they were described on the first part of this introduction. Since multi-object tracking provide key information for obstacle avoidance, tracking estimates must be not only precise but also continuous and reliable. Such quality criteria are negatively impacted when objects are imprecisely detected, partially observed or even occluded. In addition, spurious detections would lead to missed associations and false alarms. A multi-sensor object tracking framework must deal with all these challenging situations.

State-of-the-art data fusion techniques combine information from multiple sources achieving for instance an enlarged field of view using sensors covering different ranges. Different sensing strategies (e.g. vision, LIDAR, RADAR) can also reduce occlusions and missed detections. The integrity of the tracking estimates can be increased by means of multiple detection modalities and sensor redundancy. All these high-end methods are however subject and limited by intrinsic sensor errors, modelling assumptions, the complexity of associating data of heterogeneous sources and the uncertainty of estimated system parameters (e.g. calibration).

Our motivation is driven by the fact that the complexity of the environment can be efficiently reduced using contextual information. Such information is usually available in a semantic fashion. Some questions that arise are: How to efficiently integrate contextual information? Where to find an accessible and an informative source?, How to design a robust algorithm to evaluate all data together even in case of incompleteness?

Finally, it is natural to consider in the scope of this research the evaluation of developed methods. Two common ways to achieve a complete analysis of the investigated concepts are: (1) to generate synthetic emulation of data or (2) to use a public-accessed benchmark dataset. The

first choice is generally adopted when the dataset is absent. However such data is not realistic enough to achieve a reliable evaluation (e.g. noise levels, outliers, temporal misalignment of data). Full-scale datasets are composed of sensor recordings with Ground Truth annotations. It is worth noting that a dataset should ensure the independence of sensing data (perception) and Ground Truth measurements.

The investigation of the stated problems is not only intended to address the enounced issues but also to retrieve new insights about a context-aided multi-modal tracking system where all interacting functions make use the probabilistic methods. This idea might lead into a system implemented under a unique mathematical formalism.

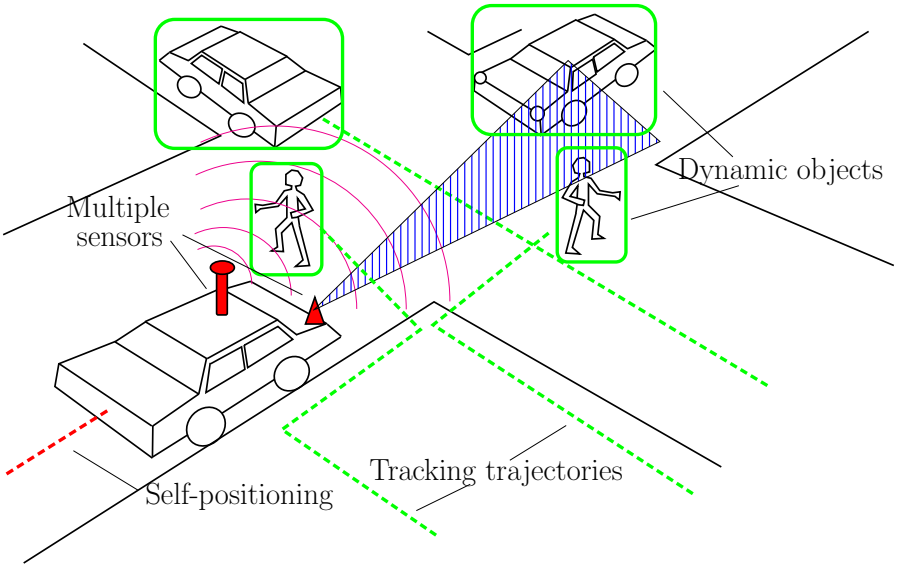


Figure 2 – The basic concept: an intelligent vehicle equipped with multiple sensors able to recognize the environment and to track dynamic objects. A self-positioning task is closely related.

Thesis structure

This thesis is composed of a detailed description of investigated methods. Chapter 1 presents a state-of-the-art survey. Then, Chapters 2, 3 and 4 respectively present in details object tracking, multi-sensor data association and contextual data fusion for intelligent vehicles. At the end, Chapter 5 and 6 describe the creation of a dataset and the experimental evaluation using ZOE platform providing a proof of concepts under full-scale scenarios. The thesis structure is illustrated in Fig. 3.

Multi-modal perception: state of art

The chapter 1 of this work, "Multi-modal perception: state of art" is a short description of multi-sensor tracking methods for autonomous vehicles. Firstly, a technological section is provided. This section lists the sensors types employed by data fusion methods for tracking. Visual sensors, radio- and laser-based devices are presented and multiple configurations reviewed. External information sources are considered as well, denoted as contextual information. The

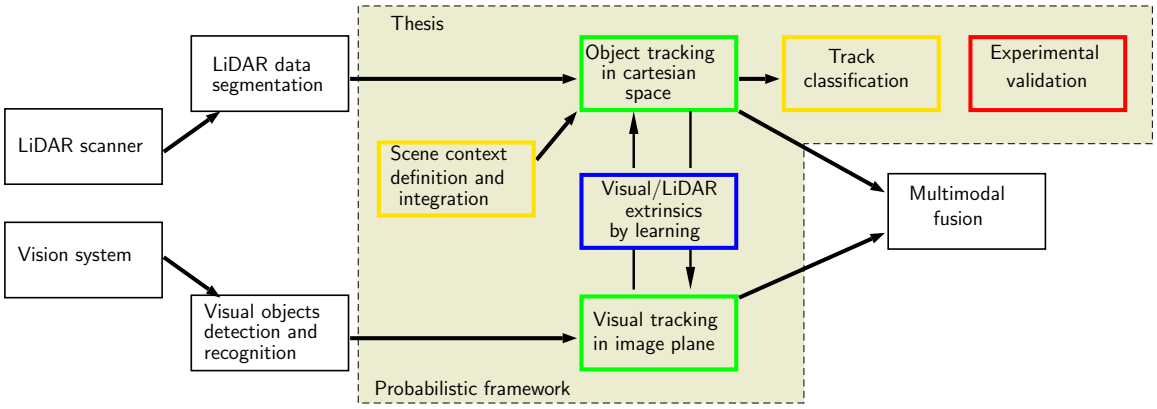


Figure 3 – Thesis structure by chapters: Green rectangles correspond to Chapter 2, Blue to Chapter 3, Yellow to 4, and Red to Chapters 5 and 6.

second section of the chapter is devoted to a methodological survey. Then, classified data fusion methods are introduced according to different levels of data representation. This chapter is aimed at explaining the context of this study and to justify the choices of developed approaches.

Object tracking

The chapter 2 explains the fundamental definitions and specifies the object tracking as a recursive process. A half of the chapter is devoted to survey tracking approaches. In the second part of the chapter, a probabilist multi-object tracking system was proposed. It is a variant of existing methods mentioned in the first half of the chapter. The tracking system is Bayes-based following a Monte-Carlo implementation. Such implementation allows for an easy contextual information integration according to the method described in chapter 4. The efficiency of the multi-object tracking system is quantified at the end of the chapter.

Multi-sensor data association

Chapter 3 addresses multi-sensor data association. A probabilistic learning method of sensor spaces association is proposed here. The main advantage of this approach is that it avoids the necessity of a calibration procedure and provides enough accuracy for LIDAR-vision applications. This chapter is a contribution to multi-modal tracking. It is important to note, the "association" term is largely used in tracking as a temporal association between detections of one tracked object. Here, only spacial association between two sensor spaces are considered. The evaluation of the association method is provided for public datasets.

Context

Chapter 4 describes a proposed method for integrating contextual information during the tracking process in order to improve accuracy and stability. The approach is based on the probabilistic representation of the tracking system and from that can be considered also as probabilistic method. An additional feature brought by this implementation is the possibility to determine if a tracked object has an attended behaviour or an unattended one. This information is crucial for safety applications. Experiments demonstrate the tracking improving in case when

the object moves according to the contextual information and the absence of the significant tracking degradation when the tracked object does not follow contextual prior. The context is represented in form of Open Street Maps annotations.

Dataset

Chapter 5 contains a detailed protocol for the creation a dataset using recordings of multiple sensors, both implemented on-board an intelligent vehicle and attached on the tracked objects. An survey of existed datasets is provided to highlight the need of the dataset with referenced multi-sensor-based Ground Truth in outdoor applications. Accurate sensors calibrations, Ground Truth labelling as well as the raw observations are included in the dataset.

Results

Chapter 6 reports evaluation results of all proposed methods on recorded dataset. Reported results confirm the conclusions about the performance of the methods on full-scale scenarios.

Chapter 1

Multi-modal perception: state of art

Contents

1.1 Introduction

The idea of multi-sensor combining is came from the nature of human perception, where he uses visual, audio, tactile, olfactory and other senses [Shimojo and Shams 2001].

For an intelligent vehicle or any other similar application, the multi-modal perception and data fusion are simpler. They does not try to understand the human brain functions, but they only model them with simple hypotheses to allow their implementation. And, at the same time, they are harder, because the "techniques" (e.g. sensor measurements, signal processing, fusion algorithms and reasoning) did not achieve the level of human data perception and fusion ability.

In this chapter the state of art of multiple sensing are presented, the common wide-spread techniques and methods are mentioned.

Firstly, some definitions are introduced.

[Hall, Liggins, and Llinas 2009] discusses the data fusion definition and finally gives a very short one:

Data fusion is the process of combining data or information in order to estimate or predict entity states.

Multi-modal perception is the process where a perceptual system combines information from more than one modality. For technical applications, the term modalities stands for the sensor sources.

Considering these two definitions, the study is intended to conceive a system able to perceive by means of multiple modalities and to combine them for estimating or predicting entity states.

This chapter addresses the questions enounce hereafter:
Which methods are more efficient to achieve the goal of multi-modal data fusion? The chapter finally focuses on some methods developed in further parts of the thesis.

Why data fusion is needed? Fused data always provide more information than each of separate modality itself. Redundant sensors can improve the dynamic target states estimation using statistical principle, since there is more independent observations. Several sensors give also an improved observability of the scene.

What are the data fusion problems? Normally, for a set of identical sensors, their physical positions, orientation, visibility can be different. Some areas of observations are perceived by only one sensor and others are perceived by multiple sensors, so the process of the target state estimation in the sensors boundary zones poses more issues about how to use information describing current target state, provided from different sources in the most optimal way.

For dissimilar sensors some additional problems appear: How to compare the sensors visibilities, precision, type of provided information in the process of fusion? If sensors give contradictory information, how to determine which is more reliable?

The State of art chapter has an aim of short review of common large-used technologies and methodologies and is not a presentation of an original research. Various compilations and classifications papers [Llinas and Hall 1998; Hall, Liggins, and Llinas 2009; Crowley 1993; Esteban et al. 2005] of data fusion was used as well as particular publications with original methods.

1.2 Technology

In this section the sensing technology used in automotive multi-modal perception is surveyed. For the external object perception, the optical, magnetic, acoustic and other devices based on remote sensing are used. Since the scene understanding is tightly coupled with the ego-localization, also the sensors based on inertial or mechanical (like Wheel Speed Sensors) principle are used [Urmson et al. 2009; Welch and Foxlin 2002]. The ego-localization with object localization in vehicle ego-oriented coordination system gives a global positioning for external objects. An example of this combination is proposed in [Sachs et al. 2008].

1.2.1 Camera

Camera is an optical passive sensing mean for recording or capturing images. For projective cameras its working principle can be summarised: light enters a closed box (*camera obscura*) trough a lens, and a light-sensitive medium records the image. The time exposition of the light-sensitive medium is controlled by a shutter mechanism. A video camera operates similarly to a still camera, but records an images series in fast succession.

Nowadays cameras have a low and still decreasing prices leading to theirs large using, including tracking purposes [Svoboda, Hug, and Gool 2002]. The multi-object tracking in civil applications today has the camera sensing as one of the base mechanisms [Sankaranarayanan, Veeraraghavan, and Chellappa 2008].

There are various camera types and corresponding models. The basic one is a pinhole camera model, where the camera aperture is a point and without lenses the light is not focused. This model can be applied even for cameras with lenses of low curvature. This model is described with more details in Sec. 5.3.2. There are a lot of applications of pinhole cameras for the moving object tracking in urban environments [H. Cho et al. 2014].

A fisheye lens is a lens providing very wide-angle of view with a strong distortion. They are used to make large, panoramic and hemispherical images. Fisheye cameras are also used in target tracking by autonomous vehicle in road applications [Held, Levinson, and Thrun 2013].

The camera with the most large, 360-degree field of view is called omnidirectional. The camera system includes additional mirrors to collect as much lights beams as possible. The autonomous driving with object tracking using omnidirectional cameras and even their stereo combination (see the further) also has its application [Vatavu, Costea, and Nedevschi 2015].

Stereo vision

Stereo vision is a perception procedure where 3D information is estimated by means of two cameras placed side by side. Two cameras provide a pair of view of a scene, like human binocular vision. By comparing such two images, the relative depth information may be extracted and represented as a disparity map. Such a map is composed as the difference in coordinates of two corresponding image points. To achieve a stereo effect two cameras outputs are processed, either in a classical computer, or there are specially manufactured stereo cameras, where the 3D information processing is inside implemented.

In Fig. 1.1a, the basic principle of stereo vision is illustrated: a point in the real world is differently observed into two frames due the disparate camera positions. The same observed point has different positions on left and right cameras projections, so one can measure the distance between its positions in projections. This distance is denoted d and is calculated as $d = d_1 + d_2$, where d_1 and d_2 are the distances between projected point and the focus. The depth value D is obtained according to the formula:

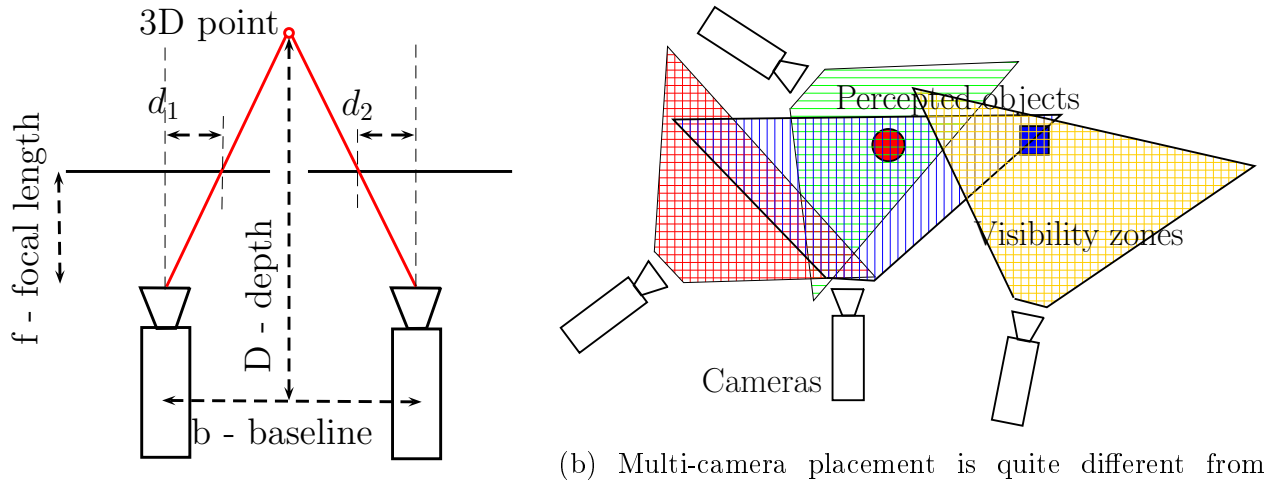
$$D = f \cdot \frac{b}{d} \tag{1.1}$$

where f is a focal length, and b is a baseline - distance between cameras.

In order to estimate the depth of a point, it must be visible from both cameras and a

processing must identify this point in both images (that is not a trivial task).

Two camera's stereo vision is cost effective and easy-to-setup vision system providing 3D (with depth) space of tracking. Stereo vision is used for multiple object tracking in traffic scenarios [Vatavu, Danescu, and Nedevschi 2015].



(a) Stereo vision basic principle. The depth is calculated from projection points disparities d_1 and d_2

(b) Multi-camera placement is quite different from stereo. Cameras observe their areas under different angles with different blind zones. On the figure the zones of visibilities are shown

Figure 1.1 – Camera combinations for the data fusion

Multi-camera perception

In contrast to the stereo vision, multi-camera perception is composed of widely separated cameras in order to obtain visual information from different viewing angles and offers a possible 3D solution [Chang and Gong 2001].

The general problem of a multi-camera perception is to determine the association of observations corresponding to the same object. Using a network of cameras needs to model the time correspondence, the overlapping, the orientation views. Such models can be obtain with calibrations techniques or by learning techniques [Heng, B. Li, and Pollefeys 2013].

Multi-focal vision is a specific example of multi-camera perception. This problem is a part of visual servoing, where robot controls its camera parameters, position and orientation based on the visual information. As in [Kühnlenz 2007; Dickmanns 2003] visual servoing can use two or more vision sensors and also be applied in autonomous vehicles vision systems.

Thermographic camera

Infra-red or thermographic camera is a passive sensing mean, forming an image like an optical camera, but using thermal (infra-red) radiation (wavelength of 14 μm) instead of visible light (wavelength 400-700 nm).

Normally, infra-red cameras are monochrome and does not distinguish wavelengths of infra-red spectre.

Multiple object tracking with thermal [Bertozzi et al. 2004] and both optical and thermal [Goubet, Katz, and Porikli 2006] cameras has its applications in urban scenarios. The

advantage of thermal cameras is the pedestrians detection, because frequently they are warmer than the environment.

1.2.2 RADAR

RADAR is an acronym for RAdio Detection And Ranging. RADAR is a system designed for object-detection, using radio waves for determining the object properties, like range, velocity, angle, etc. A RADAR system consists in a transmitter to operate electromagnetic waves in the radio or microwaves spectrum, an emitting and receiving antennas, a receiver and a processor to determine the object properties.

A transmitter emits radio waves in predetermined directions. After a contact with an object, signals are reflected or scattered in many directions. RADAR signals are better reflected by materials of considerable electrical conductivity - by most metals, seawater and wet ground. Those signals, reflected back to the transmitter, are desirable. The principle can also be employed to determine object motion based on the Doppler effect.

Since the radio waves are weakly absorbed by the medium, the RADAR can detect objects at long ranges.

In multi-object automotive perception, the RADAR has been largely applied [Darms, Rybski, and Urmson 2008]. The RADAR data fusion with other sensors in multi-object perception has been extensively a subject of research [Liu, Sparbert, and Stiller 2008; Urmson et al. 2009].

Compared with LIDAR, RADAR has some disadvantages: RADAR can be perturbed with multiple reflections, it is less accurate than the reflection of straight laser beam. The advantage of the RADAR is its large angle of view.

1.2.3 LIDAR

LIDAR or laser ranging, is an acronym for LIght Detection And Ranging. This is a remote sensing technology emitting intense, focused light beams and measuring the time of flight for reflections to be detected by the sensor. This information is used to determine distances (ranges) to objects [Oceanic and Center. 2012].

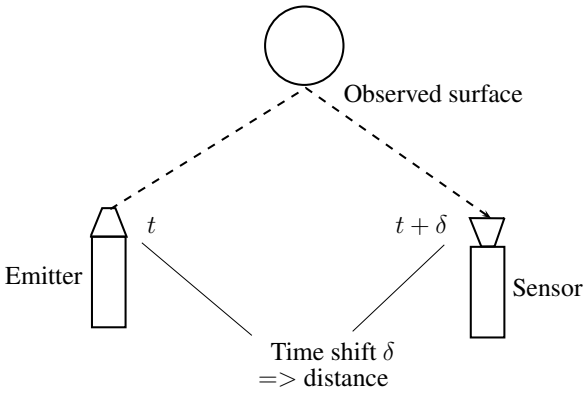
The LIDAR technology relies on the same principles as the RADAR, but uses much shorter wavelength (ultraviolet, visible or near infra-red range) than RADAR's radio waves. Given the fact that sensors are not able to detect objects smaller than the used wavelength, LIDAR is able to detect surfaces with finer resolution than RADAR.

The LIDAR measurements are generally "point-wise", it computes the distance to only one point. To have a complete range map it is required to have multiple measurements. For this end a rotation mirror is used for re-orient the laser emission. Moreover, there is a need of a delay between two measurements, which makes the observed scene not strictly rigid.

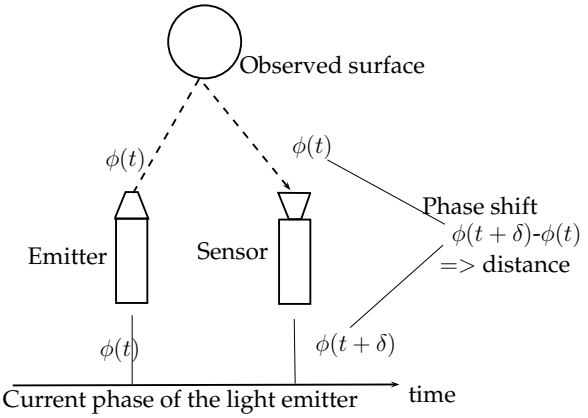
LIDAR system include a laser range finder. The finder orientation is changing by a rotating mirror. The laser is scanned around the scene and converts the range measurements into 2D plane each layer at specified angle intervals. The 3D image is made as soon as layers are in differently oriented planes. The basic principle is shown on Fig. 1.2a.

LIDAR has been used for multi-object perception as a single tool [Himmelsbach et al. 2008;

Nobili et al. 2015] and in a sensing-cooperative architecture with other sensors [Premebida, Monteiro, et al. 2007]. The main advantage of the LIDAR system in comparison to the visual optical instruments is its high resolution at large distances from the sensor.



(a) LIDAR basic principle. Point-wise measurements of range are achieved measuring the time delay between emission and reception



(b) ToF camera basic principle. The range is calculated by measuring phase delay for each pixel

Figure 1.2 – Light-based range sensors

1.2.4 Time-of-Flight

Time-of-Flight (ToF) technology is based on measuring the time required by an emitted light to travel to an object, and return back as a reflection. This is the principle of LIDAR presented in Sec. 1.2.3, but implemented in standard CMOS (active-pixel sensor) or CCD (charge-coupled device) technology, it is denoted ToF camera [Kolb et al. 2010].

The device is composed of two parts: a near infra-red light source and an optical sensor. The source emits a phase-modulated signal. The optical sensor in addition to the intensity of the reflected light is able to capture the phase of the received signal. The modulation of the signal is synchronized between the light source and the sensor. Pixels values represent the depth that is computed by comparing the phases modulation at the perception time moment with the phase of the initial signal. The illustrated schema is shown on Fig. 1.2b

ToF sensors provide dense depth measurements on a scene at high frame rates (up to 30 Hz). A disadvantage of the modern ToF implementations is a limited range, working well in indoor use and not very well outdoor [Luettel, Himmelsbach, and Wuensche 2012]. Still, there are applications of multiple object tracking using ToF cameras installed on automotive robots even for outdoor scenarios [Jafari, Mitzel, and Leibe 2014]. In [Hsu et al. 2006], a ToF cameras-based safety application for intelligent vehicles perception.

1.2.5 Global Positioning System

The Global Positioning System (GPS) is a space-based navigation system. GPS provides location and time information anywhere on the Earth or near it in all weather conditions, having an unobstructed line of sight to 4 or more GPS satellites.

Under the GPS is considered the global navigation satellite system (GNSS) including also GLONASS (GLObal NAvigation Satellite System) and others.

The base definitions and descriptions in this section are inspired by [Kaplan 2005].

Pseudo-distance GPS

GPS uses the concept of ToA (Time of Arrival) measurements to determine the receiver position. This concept entails measuring the time needed for an emitter-transmitted signal (e.g. satellite) at a known location to reach the receiver. By measuring the propagation time from multiple emitters at always known locations, the receiver can estimate its position.

To calculate the receiver location, the trilateration is used. The trilateration in geometry is the process of determining the point location based on distance measurements and geometry of triangles, circles and spheres. In Fig. 1.3 the basic principle is shown. Each satellite provides a line in the Earth surface where points are at given equal distance from the satellite. Tree lines are enough to achieve positioning. For calculating all these distances from satellite, time synchronization is required. It is worth noting that satellites are able to provide precise atomic clocks updates, however the receiver is that precise. To cope with receiver clock error, the signal from a fourth satellite is used to constraint the following equation system:

$$b_i = c(\Delta t_i + t_c) = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \tag{1.2}$$

where i is the ith satellite, b_i is a distance between receiver and satellite i, c denotes light speed, Δt_i is the elapse time of signal propagation from satellite to receiver, t_c is a correction of the receiver clock, x_i,y_i,z_i are coordinates of i_th satellite, x,y,z are coordinates of receiver. With 4 unknown, 4 equations are necessary to solve.

Atmospheric conditions add independent errors to satellite signals. To reduce effects of such a phenomenon, more satellites are taken into account. Thus, the positioning precision depends on the number of visible satellites, the more information, the more precise it is.

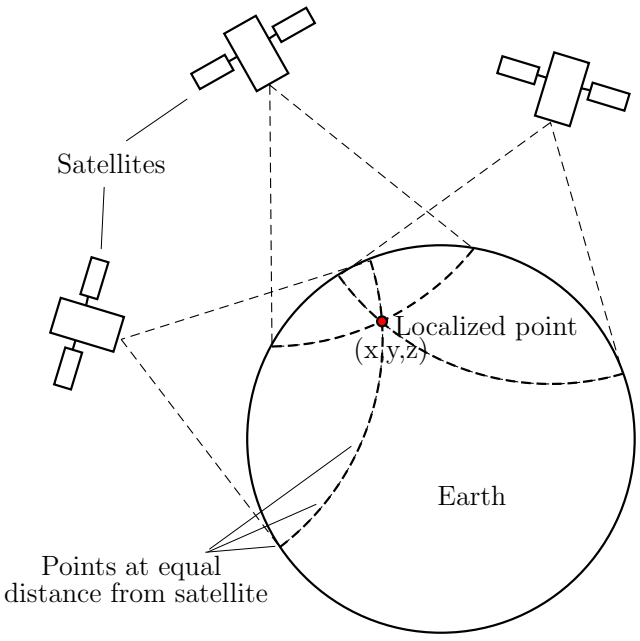


Figure 1.3 – Each satellite distance gives a line of position on the Earth. Tree lines are enough to resolve a single position on the Earth

Two types of messages are transmitted from satellites:

— "Raw measurements", meaning code phase and carrier Doppler phase (or frequency). These data plus timestamps are enough to calculate pseudo-ranges between receiver and satellites.

— Navigation data: Ephemeris parameters about satellites trajectories are received by GPS satellites from ground referenced antennas and then sent back to user receivers.

These two message types are sufficient for positioning. There are various modifications of the described positioning schema to enhance the fix precision, such as SBAS, DGPS, RTK, and other. Some of them are described in a further section.

Satellite-Based Augmentation System (SBAS)

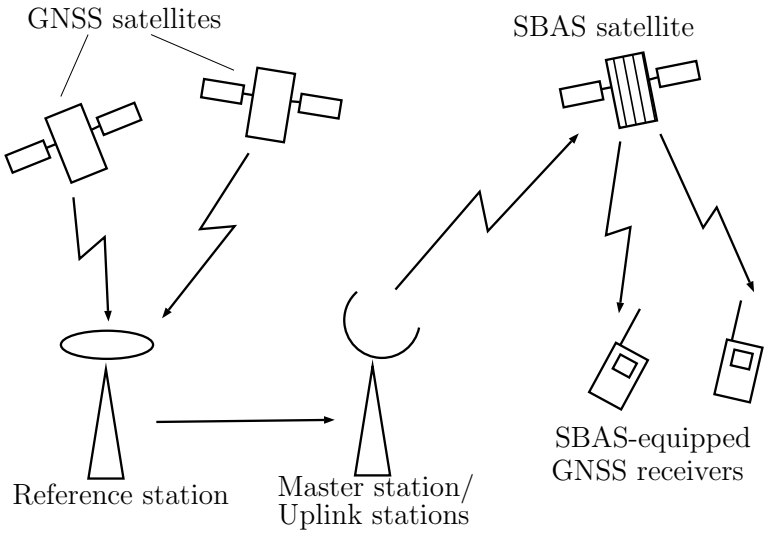


Figure 1.4 – SBAS systems common schema

Satellite-Based Augmentation System (SBAS) is composed of widely dispersed reference stations monitoring and gathering data about GPS satellites.

The key elements of SBAS are ground referenced stations, distributed in a service area, SBAS master station and SBAS satellites. Referenced stations receive GPS signals from GPS satellites, calculate the area corrections using known locations of the stations and send these corrections to the master station, which transmits them to SBAS satellites. SBAS satellites signals are broadcasted to GPS receivers. This schema is illustrated in Fig. 1.4.

SBAS system not only enhance the positioning precision through the transmission of corrections, but also helps to quickly detect satellite signal errors and send alerts to receivers to do not use unreliable satellite information. The SBAS satellite can also be used as GPS satellite to provide additional ranging signal.

There are several SBAS systems, including US Wide Area Augmentation System (WAAS), European Geostationary Navigation Overlay Service (EGNOS), Japanese MTSAT Satellite Based Augmentation Navigation System (MSAS), Indian GPS-Aided GEO Augmented Navigation System (GAGAN), Russian System for Differential Corrections and Monitoring (SDCM) and Chinese Satellite Navigation Augmentation System (SNAS).

Ground Based Augmentation System (GBAS) uses a very high frequency radio link and also provides differential corrections and satellite integrity monitoring. GBAS covers a small

area and is used where high accuracy, availability and integrity are required, for example in airports.

Some approaches where SBAS is employed for autonomous vehicle positioning [Dixon 2006; Toledo-Moreo et al. 2007].

Differential GPS

The Differential GPS (DGPS) uses one or more referenced stations at known locations, equipped with their own GPS receivers. The fixed GPS receiver is denoted as a base station providing positioning corrections to a mobile receiver. DGPS is considered as a high accuracy positioning system with conventional surveying techniques.

The base station compares its mean surveyed position with the position calculated from currently received satellite messages. The difference between those positions is considered as a result of satellite ephemeris and clock errors, but more frequently as an atmospheric delay.

The base station provides information messages to other receivers via a data link, containing corrections to raw and user's pseudo-range measurements and clock corrections provided by satellites. The base station can also provide ephemeris data or data to replace the broadcast clock and ephemeris information. Instead of corrections, base stations can also provide raw measurements, pseudo-ranges and carrier phase.

Using those corrections the receiver station may enhance its pseudo-ranges, time and ephemeris respectively.

The absolute accuracy of the calculated receiver position depends on the absolute accuracy of the base station itself.

The principle is based on the fact that GPS satellites orbit is high above the Earth and the propagation paths from the satellite to the base and rover station are similar. DGPS principle can increase positioning accuracy up to 10 cm with base and rover separated to tens of kilometres.

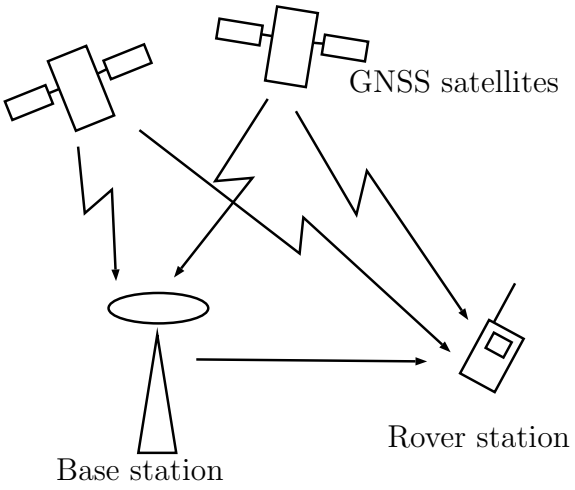
In multi-sensor data fusion applications for object tracking in autonomous vehicles the DGPS is used for the rover positioning [Chavez-Garcia 2014; Chumerin and Hulle 2008].

Real Time Kinematic (RTK), RTK with inertial center

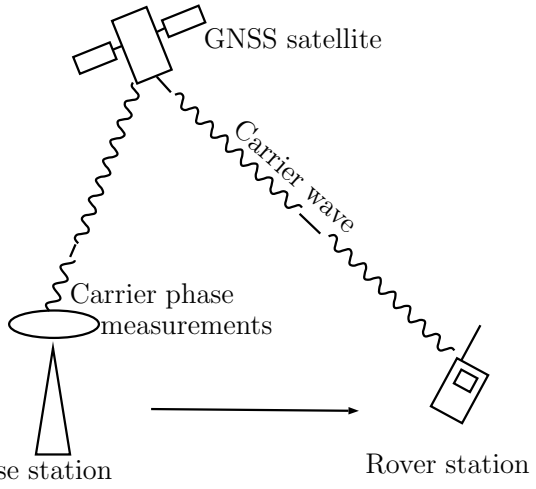
DGPS and single-receiver positioning are code-based, i.e. they use the time codes in messages to synchronize receiving time and then calculate pseudo-ranges. Another approach, the Real Time Kinematic (RTK) utilize the L1 (1575.42 MHz) carrier-phase measurement. Code and carrier-phase measurements are available from each satellite. Dual frequency GPS receivers are able to employ such measurements for both the L1 and L2 (1227.60 MHz) frequencies.

The common principle of RTK is the range calculation from the number of carrier cycles between receiver station and satellite. The range is calculated as number of carrier cycles multiplied by carrier wavelength. The process to determine the number of carrier cycles is called ambiguity resolution.

While using this signal the possible improvement is very high (pseudo-range error might be as low as 2 millimeters) if one continues to assume a 1% accuracy in locking. For L1 band the



(a) The base station compares the surveyed position to the position calculated from the satellite ranges. After that, the base sends this difference error to the rover station



(b) The number of carrier cycles from the satellite to the equipment is determined and used to calculate the range. Base station, as in DGPS, sends corrections to the rover station

Figure 1.5 – Methods using base station with known locations as reference

wavelength is about 19 cm. That means one percent error in L1 carrier phase brings 1.9 mm error in the initial positioning.

In practice, as in DGPS, RTK systems use a single base station receiver, that re-broadcasts the phase of the carrier it observes and the mobile receiver compares its phase measurements with those received from the base station. The base station is used to eliminate satellite clock errors, ephemerides errors and delays produced by ionosphere and troposphere.

RTK method is considered as the most precise GPS positioning without inertial tools. In multi-modal object-tracking by autonomous vehicles, the positioning with RTK has its applications [Geiger, Lenz, Stiller, et al. 2013].

Inertial navigation system is a navigation system equipped with motion sensors (accelerometers) and rotation sensors (gyroscopes) to calculate the position, orientation and velocity of the object during dead reckoning, e.g. GPS outages. In some GPS devices, motion sensors are integrated and they also helps to increase the positioning precision. There are also external connections between GPS devices and motion sensors [Scherzinger 2000; Schall et al. 2009].

Data post-processing

For some applications, GPS corrections are not required in real-time. In these cases, raw GPS satellite measurements are collected for post-processing. This processing is very useful, as it does not require real-time transmission of differential correction messages. This simplifies the system configuration and also eliminates possible message loss and latency during such a transmission.

Post-processing generally results in a more accurate, comprehensive solution than in real-time.

1.2.6 External sources

Additionally to the vehicle on-board sensors, there are also intelligent vehicle systems designed to communicate information with other platforms (vehicle to vehicle, V2V) [Derder, Moussaoui, and Boualouache 2015], with infrastructure (V2I and I2V) [Derder and Moussaoui 2014], and hybrid systems [J. Miller 2008]. Such architectures usually provide complementary exteroceptive information of the vehicle surroundings that cannot be observed from on-boarded sensors. The use of external sources in intelligent vehicles' tasks can be considered also as internal and external data fusion. It serves for the traffic regulation, target tracking, incident warning, etc.

Environment maps can be employed as an external information source [Quddus, Ochieng, and Noland 2007]. In Sec. 4 the applied method takes advantage of a map to improve multi-object tracking.

1.3 Methodology

1.3.1 Data representation for fusion

Data fusion is a large research subject, covering several areas, so today there is no common and unique fully covered classification of data fusion methodology [Castanedo 2013]. A classification based on different abstraction levels is presented [Luo, Yih, and Su 2002]. That classification can provide information about applications purposes. For example, signal level fusion as the most primitive can be used in real-time or be a step to higher levels of data representation. Pixel level fusion improves the performance in image processing - the domain is large enough to mark out a separate level. Feature and symbol levels give additional forms for data representation. Also, the levels classify the type of provided information, determine the degree of required sensor registration, and separate methods and means used to increase the data fusion impact.

According to this classification, the process of perception is structured on the next levels:

1. Signal level - here the information is represented as signals acquired from the sensors
2. Pixel level - in applications using image data, this level can be used to improve image processing. Sometimes, signal and pixel levels are coupled in data level.
3. Characteristic (feature) level - employs features extracted from images or signals
4. Symbol (decision) level - information here is represented as high-level symbols

The characteristics of the levels are given in Tab. 1.1.

Data fusion in image and spatial applications can be classified according to fusion objectives and input data properties. In Fig. 1.6 the corresponding taxonomy is shown. In this work, after the state of art survey, the General Data Fusion Problem and Spatial Data Fusion are studied in detail, particularly in Chap. 3 and in Chap. 4. The authors of [Crowley 1993] have formulated the following principles for integrating perceptual information:

Principle 1) Primitives in the world model should be expressed as a set of properties.

Data fusion is an association of properties describing some state of a world part. In numerical representation, properties are listed as estimations with their uncertainty. At symbolic level, the

Characteristics	Signal level	Pixel level	Feature level	Symbol level
Representation level of information	Low	Low	Medium	High
Type of sensory information	Multi-dimensional signal	Multiple images	Features extracted from signals/images	Decision logic from signals/image
Model of sensory information	Random variable with noise	Random process across the pixel	Non-invariant form of features	Symbol with degree of uncertainty

Table 1.1 – Characteristics of data fusion levels according [Esteban et al. 2005]

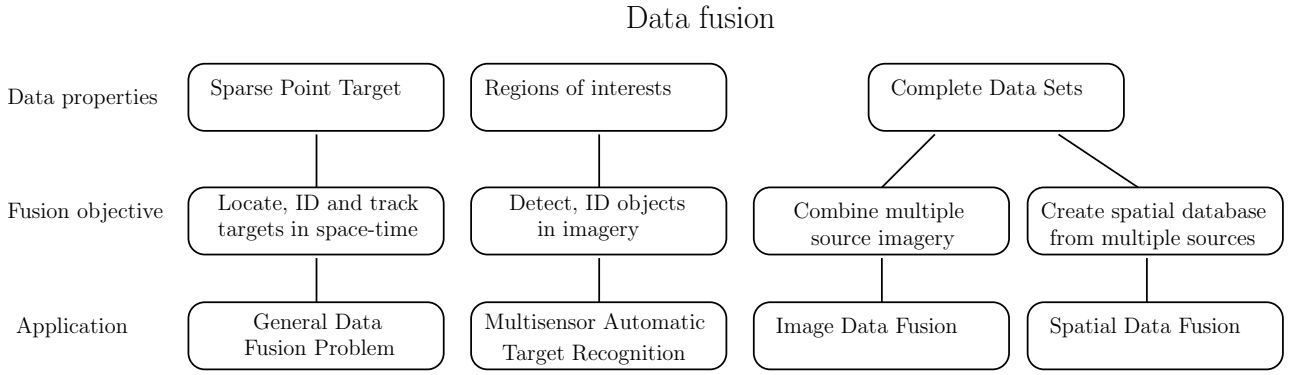


Figure 1.6 – Image and spatial data fusion application taxonomy [Hall, Liggins, and Llinas 2009]

properties values belong to a finite vocabulary, for example, hypotheses about object position can contain a finite number of regions. In that case an object position can be represented with a region index.

Principle 2) Observation and Model should be expressed in a common coordinate system.

To evaluate a perceptible entity's model using observations, the information from observations must be transformable to the coordinate system of the model. Normally, observation coordinate systems are defined by sensors. Thus, it is necessary to adapt the corresponding models to make the fusion possible. For the adaptation some additional information is needed. For example, an extrinsic calibration between sensors provide a type of information. From other side, probabilistic data fusion approaches need supplementary information about statistical relations between observations from different sensors.

Signal level

At the signal level, the information is represented as signals from the sensors. To apply data fusion techniques, signals must be represented in forms close to features or symbols. For instance, audio and video-sensing can give input raw signals in form of spectral vectors with easy computable statistics. From this point of view, the data fusion problem become probabilistic [Gustafsson 2005].

At higher levels of data representation, to fuse information about an object observed by multiple sensors, a region of interest or some extraction processing from raw data is needed.

To carry out a signal level fusion, the raw signal must be considered as an extracted target or a region of interest.

Pixel level

The image fusion combines multi-source imagery using image processing techniques. The image fusion is applied to achieve different purposes [Pohl and Genderen 1998]:

— Image filtering
— Image Mosaicking to achieve a larger view
— 3D reconstruction from two 2D viewpoints
— Data fusion for classification. For example, thermal+color pixel characteristics
— Change detection over time using multi-temporal data
— Reconstructing missing information
— Corrupted data identification

The pixel level fusion operates with a slightly different data from identical or not very distinct sources. Some well-known methods performing image data fusion at a pixel level are: Intensity–Hue–Saturation (IHS) transform based image fusion [Wen 2011], Principal Component Analysis (PCA) based image fusion [J. W. Davis and V. Sharma 2007], Pair-wise spatial frequency matching [X. Li, Larson, and Hanjalic 2015]) and transform domain fusion: Pyramid-based transforms [S. Zheng 2010], Wavelet transform image fusion [Z. Wang et al. 2009]. Transform techniques show a better performance in spatial and spectral quality of the fused image. It is interesting that PCA, Wavelet transform and some other methods used for image fusion can be also applied on feature extraction to detect objects of some defined classes in images [Sun, Bebis, and R. Miller 2006].

Feature level

A Feature is composed of individual measurable properties of the observed phenomena. For instance in the problem of dynamic object perception, the features can be the representations of objects and their detections enriched by multiple sensors. A feature representation can be defined as points in some area, their velocities and other structure characteristics. Frequently, an extracted feature is represented as a numerical vector, used in further processing, like classification or fusion [Caron et al. 2014]. The feature level data fusion is the most usual in the following processing: data alignment, association, identification.

One of the most simple data fusion at feature level is a concatenation of features, represented as vectors [Hassan, Shroff, and Agarwal 2015; H. Cho et al. 2014]. This approach increases the state space of objects to detect, to track or to classify. A concatenated features has a higher dimension and can be easily distinguished in the representation space. This reduces the error for classification tasks, but at the same time, it may highly increase the computation cost, which is crucial for real-time applications.

Methods based on the choice between sources use features, because a feature can contain information about its certitude or about the quantity of the information that the feature brings. For example, in [Pramanik and Bhattacharjee 2012] the features are statistical moments calculated on multiple characteristics of an image. For a couple of sources, the features sets

are extracted. Then an identification of "salience" features processing is applied. The salient features entail a decision map, helping to form a fused image.

A feature histogram vote-based method is proposed in [M. Wang et al. 2010]. Here the classical Histogram of Oriented Gradients (HOG) [Dalal and Triggs 2005] features extracted from image representation of visual and thermal cameras are fused into one common histogram.

A Principal Components Analysis (PCA) can reduce the concatenated feature as it can extract and keep the most important components of feature [Khairdoost, Monadjemi, and Jamshidi 2013]. PCA is also used in image object detection to extract correlations, even among pixels.

In [Cramer, Scheunert, and Wanielik 2003], the logical and geometrical relations between sensors and corresponding features are known from a model describing vehicles and pedestrians. In that case, a common feature is constructed from separate features using linear (or more sophisticated) transformations.

Symbol level

One of the data fusion approach at symbol (decision) level is based on fuzzy logic, where the decision results of different sensors are combined using many-valued logic operators. Fuzzy logic fusion has applied, for example in [Ghahroudi and Fasih 2007].

Weighted sum is another simple model in multi-criteria decision analysis. Here the resulted response is calculated as weighted sum of multiple sources' logical decisions. To this end, weights are normalized by the likelihood of uncertainty assigned to the sources. Weighted decisions are easy to be implemented and in some kind universal, so they find their applications, frequently in a complex classifier composition from weak classifiers. Such classifiers are applied in data fusion methods for multi-sensor intelligent vehicles road applications [Premebida, Monteiro, et al. 2007; Vu, Aycard, and Tango 2014]. When using the weighted sum method, one must estimate weights for the information sources. Most of the solutions are concentrated near anchor points, and not in the concave region, because weighted sum method's solutions are distributed not uniformly and can not be found in non-convex regions.

Classical or Frequentist inference (also called frequentist statistics) is a type of statistical inference that draws conclusions from sample data by emphasizing the frequency or proportion of the data. In data fusion methods, the classical inference-based fused decision comes from statistic decisions of fused sources [D. Cox and Mayo 2010]. The frequentist interpretation of the probability is slightly different from the Bayes' one, and since that, the classical inference differs from the Bayes approach. Bayes approach becomes much more popular in data fusion task [Fienberg 2006].

1.3.2 Fusion formalism

Bayes methods

The Bayesian data fusion method is one of the most frequently used at symbolic level, but also at other levels too. It is largely applied in object tracking by intelligent vehicle's sensors for urban scenarios [Pangop et al. 2008; Stiller, León, and Kruse 2011; Premebida, Peixoto,

and U. Nunes 2006; Vasic and Martinoli 2015]. This methodology can be detailed in application to system state estimation using multiple sensor observations. The Bayesian method has a mathematical background. The Bayesian rule facilitates representing and taking into account parameters and model uncertainties. It provides a natural, intuitively clear probabilistic combining of prior information and interpretable answers. It can fuse past information with new observations by using old posterior as new prior. It obeys the maximum likelihood principle. A difficulty is to model the *a priori* information through known distributions, such as Gaussian, Poisson,... *A posteriori* estimation is naturally dependent and inadequate *a priori* models can be a non-detectable source of errors. In models with a high number of parameters its computational cost cannot be neglected.

Let the state system to be estimated, $\mathbf{x}_k$, at discrete time k, based on all previous sensor measurements from all m sensors $\mathbf{z}_{1:k}^{1:m}$. The task of the state estimation in probabilistic terms is the computation of the posterior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k}^{1:m})$. When applying Bayes theorem, the problem can be formulated as follows:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{1:k}^{1:m}) &= p(\mathbf{x}_k|\mathbf{z}_k^{1:m}, \mathbf{z}_{1:k-1}^{1:m}) \\ &= \frac{p(\mathbf{z}_k^{1:m}|\mathbf{x}_k, \mathbf{z}_{1:k-1}^{1:m}) p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^{1:m})}{p(\mathbf{z}_k^{1:m}|\mathbf{z}_{1:k-1}^{1:m})} \end{aligned} \tag{1.3}$$

Assuming that given the state $\mathbf{x}_k$ the measurement at i^{th} sensor is independent of the measurements from other sensors and that the current state $\mathbf{x}_k$ includes all information for likelihood evaluation, then one can drop the dependency of current measurement of the i^{th} sensor $\mathbf{z}_k^i$ from all previous measurements of all sensors $\mathbf{z}_{1:k-1}^{1:m}$:

$$\begin{aligned} p(\mathbf{z}_k^{1:m}|\mathbf{x}_k, \mathbf{z}_{1:k-1}^{1:m}) &= \prod_{i=1}^{m} p(\mathbf{z}_k^i|\mathbf{x}_k, \mathbf{z}_{1:k-1}^{1:m}) \\ &= \prod_{i=1}^{m} p(\mathbf{z}_k^i|\mathbf{x}_k) \end{aligned} \tag{1.4}$$

Based on Eq. 1.4 three different data fusion strategies can be adopted:

1. If sensors report only their measurements modelled in a probabilistic manner, like a likelihood or a sensor model, then the global estimation of the system state is updated by fusing likelihoods only:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}^{1:m}) \propto p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^{1:m}) \prod_{i=1}^{m} p(\mathbf{z}_k^i|\mathbf{x}_k) \tag{1.5}$$

   where $p(\mathbf{z}_k^{1:m}|\mathbf{z}_{1:k-1}^{1:m})$ is omitted as just normalization coefficient for calculated posterior. This is called **centralized independent likelihood fusion**

2. Alternatively, modalities of own local system state can be estimated based only on local observations. Using Bayesian rule applying on the likelihood $p(\mathbf{z}_k^i|\mathbf{x}_k)$ from Eq. 1.5, the follow expressions are derived:

$$p(\mathbf{z}_k^i|\mathbf{x}_k) \propto \frac{p(\mathbf{x}_k|\mathbf{z}_{1:k}^i)}{p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^i)} \tag{1.6}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}^{1:m}) \propto p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^{1:m}) \prod_{i=1}^{m} \frac{p(\mathbf{x}_k|\mathbf{z}_{1:k}^{i})}{p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^{i})} \tag{1.7}$$

This is called the **hierarchical fusion without feedback**.

3. Finally, a global prediction based on all i sensor measurements, can serve as local prior. Such prediction is stated below:

$$p(\mathbf{z}_k^i|\mathbf{x}_k) \propto \frac{p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^{1:m}, \mathbf{z}_k^i)}{p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^{1:m})} \tag{1.8}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}^{1:m}) \propto p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^{1:m}) \prod_{i=1}^{m} \frac{p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^{1:m}, \mathbf{z}_k^i)}{p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^{1:m})} \tag{1.9}$$

This is denoted **hierarchical fusion with feedback**.

A more detailed and specialized research on Bayesian methods can be found on [Markovic and Petrovic 2014; Abdulhafiz and Khamis 2013].

Dempster–Shafer method

The Dempster-Shafer method may be interpreted as a generalization of Bayesian theory for multi-hypotheses processing [Shafer 1976; Smets 1988; H. Wu 2004; B. Ma 2001]. In a Dempster-Shafer reasoning system, the frame of discernment T is composed by the possible basic hypotheses who are not dividable and mutually exclusive. The system inference space is the power set Θ of T. For a discernment $T = \{A, B, C\}$ the inference is $\Theta = \{\{A\}, \{B\}, \{C\}, \{A, B\}, \{B, C\}, \{A, C\}, \{A, B, C\}, \emptyset\}$, where hypotheses in braces means a hypothesis in which one of the component is valid.

With the frame of discernment T and the possible hypothesis Θ defined, belief (bel) can be assigned over Θ. Like in probability, the total belief equals 1. Each sensor S_i reports its observation by assigning beliefs. The basic belief assignment is called mass function m_i of S_i. The assignment is based on the observed evidence E, supporting the belief. For a hypothesis H the belief is expressed as follows:

$$bel_i(H) = \sum_{E_k \subseteq H} m_i(E_k) \tag{1.10}$$

Another characteristic is plausibility (pl) of hypothesis H, including all the observed evidence objects that do not stand against H:

$$pl_i(H) = \sum_{E_k \cap H \neq \emptyset} m_i(E_k) = 1 - bel_i(\bar{H}) \tag{1.11}$$

In Dempster-Shafer system, the belief and plausibility form a confidence interval $[bel_i(H), pl_i(H)]$ for a hypothesis, that represents the measured belief about the hypothesis. The confidence interval is an interval, where the true probability lies with a certain confidence.

When one speaks about sensor's data fusion, there are rules which combine sensor S_i's observation m_i and sensor S_j's observation m_j such as in the close-world assumption model [Smets 1994]:

$$m_i \oplus m_j(A) = \frac{\sum_{A_k \cap A_{k'} = A} m_i(A_k) m_j(A_{k'})}{1 - \sum_{A_l \cap A_{l'} = \emptyset} m_i(A_l) m_j(A_{l'})} \tag{1.12}$$

where A_k,$A_{k'}$,A_l,$A_{l'}$ are sets from the discernment and A is a hypothesis. There are other fusion models, such as in open-world assumption [Smets 1994], in cautious rule [Denoeux 2006], etc. [McKeever and Ye 2013]

The Dempster-Shafer method can update *a priori* estimation with new observations for *a posteriori* estimations, like it was demonstrated for Bayesian inference. The Dempster-Shafer method relaxes the Bayesian restriction on mutually exclusive hypotheses. Belief theory can be used both for the tracking purposes [Mourllion et al. 2005] and multi-sensor data fusion purposes [Klausne, Tengg, and Rinner 2007] in intelligent vehicles applications.

1.3.3 Multi-sensor data association

In the previous section it was mentioned that the model and observations must be expressed in a common coordinate system. For different sensors their coordinate systems also differ. There are two possible solutions to make a data fusion in this case:

1. To have a possibility to transform observation of one sensor into the coordinate system of the other
2. To introduce a common coordinate space and also have the possibility to transform observations there

The methodology describes the transformation which are necessary to carry out the coordinate system changing. I present the state of art in this domain in two groups of methods classified as "multi-sensor calibration" and "association by learning".

Multi-sensor calibration

Sensors of different nature (and different space of detections) perform their measurements independently. Calibration serves to know whether two measurements originate from the same object or less generally: from the same physical position. To find these correspondences, standard algorithms like Detection And Tracking of Moving Objects (DATMO) usually make use of a calibration procedure which allows to transform measurements of one sensor into the reference frame of the other. Such transformations are often quite sensitive [Rodriguez, Fremont, and Bonnifait 2008] to the used measurement models (e.g., pinhole model for camera) and calibration parameters. Moreover, because of the nature of the measured quantities, sometimes a bijective transformation does not even exist. This is for example the case when transforming 2D camera image points into a 3D coordinate system of a LIDAR device. State-of-the-art approaches for intelligent vehicles such as [H. Cho et al. 2014; Rodriguez, Fremont, Bonnifait, and Cherfaoui 2011] rely on the explicit need of a common frame where all sensors observations can be referenced (i.e. data alignment). This assumption greatly simplifies the association problem of multiple data sources (e.g. LIDAR, RADAR, vision). However, in practice, a calibration

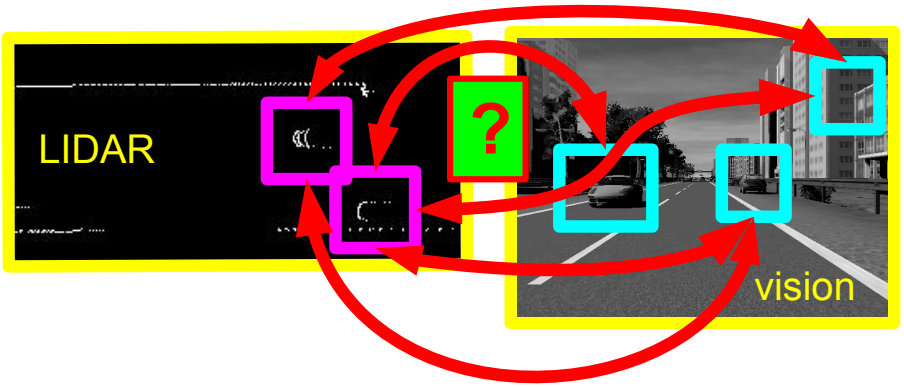


Figure 1.7 – Illustration of the multisensory correspondence problem: LIDAR (left) and visual (right) measurements, e.g., provided by independent object detection algorithms, "live" in completely different spaces and are thus very difficult to associate without applying prior knowledge.

procedure is required in order to precisely determine all sensors rigid-body transformations (i.e. extrinsic parameters) into the reference frame and their uncertainties.

As an example of external different sensor's calibration method, one can cite [Fremont, Sergio Alberto Rodriguez Florez, and Bonnifait 2012] where linear transformations between sensors' relative positions is estimated using sets of captured images and LIDAR points cloud corresponding to circular targets for some various positions.

To realize the previous calibration, intrinsic camera parameters must be also estimated with another calibration technique [Zhang 1999] requiring several images of a chessboard captured from a camera under different angles.

Recent works on the 3D sensor calibration have considerably simplified the procedure for determining the relative position of sensors using a set of natural features [Scaramuzza, Harati, and Siegwart 2007] or using a single observation of a set of calibration patterns (covering different distances and orientations of the multi sensors field of view) [Geiger, Moosmann, et al. 2012].

Automatic calibration approaches can also infer the extrinsic parameters by the means of an optimization framework which registers sensors data in a common space (typically 2D/3D Cartesian space). Recently in [Pandey et al. 2014; Levinson and Thrun 2013] and [Napier, Corke, and Newman 2013], online strategies were proposed to achieve data registration between a vision system and a ranging sensor by optimizing the extrinsic parameters using a mutual information criterion about the sensing sources.

Association by learning

In [K. Kim and L. S. Davis 2006] the full camera calibration is not needed, only a partial calibration for ground plane homography. The partially calibrated cameras can project a detected object as a line on a top-view ground plane. The intersected lines gives an object position in a common tracking space, while its associations with cameras are still available.

The article [Javed et al. 2003] proposes a method of object correspondence between non-overlapping cameras using statistical learning approach based on observations by non-inter-calibrated cameras. The learning method proposes a probabilistic space-time model of lost

object waiting: when a tracked pedestrian walks out of one camera visibility area, the other camera is waiting a new pedestrian appearance. If the time of arriving is consistent to the model, the new appeared pedestrian in second camera is associated with the lost pedestrian from the first camera.

In [Cheikh et al. 2012] a method of object association based on image feature distance is proposed. The approach don't need any camera calibration nor inter-space learning, but has strict restriction about tracked objects features. Particularly, the normalized color histograms are calculated for inter-view association.

As an alternative to the classical approaches, the method described in Chap. 3 is intended to perform multi-sensor data alignment through a probabilistic learning based framework. This approach not only provides a data alignment solution but also models the probability accorded to the observation transfer. Moreover, this method can provide an integrity measure of the data alignment using extrinsic parameters in a cross-validation scheme.

Chapter 2

Object tracking

Contents

2.1 Introduction

Fundamental definitions of this part are listed below.

Object is a dynamic entity existing in the real world, detectable by sensors and that can be modelled.

Detection is either a process of the information extraction about an object from a larger stream of information at a given time moment; or a result of the process as momentary object representation.

Tracking is the process of estimation of the moving object's or multiple objects' locations (or other data) composing the trajectory.

Tracked object is the object been tracked. In this work the tracking uses detections, which arrive in consecutive time moments.

Track state - instantaneous estimation of the tracked object location and other characteristics. The state may be defined differently according to applications. In the approaches used further, the state includes its coordinates in the corresponding space. The direction and movement speed, as well as object's size parameters may also be included in the state. In the taxonomy of tracking methods shown at Fig. 2.1 that type of tracking is denoted point tracking. As a result, tracking provides an estimation of current object's position using the observed positions of the past and external observations.

Track is a result of the tracking process, i.e. the set of time-ordered states of the tracked object.

A taxonomy of tracking methods is represented on Fig. 2.1. This figure illustrates a global classification of state of the art of multi-object tracking methods [Parekh, Thakore, and Jaliya 2014; Yilmaz, Javed, and Shah 2006]. Probabilistic point tracking is focused since it is related to the methods considered in this work. The probabilistic methods have a developed mathematical theoretical base, have wide-spread working applications and are generally common, i.e. can be used in different systems without any specified restrictions.

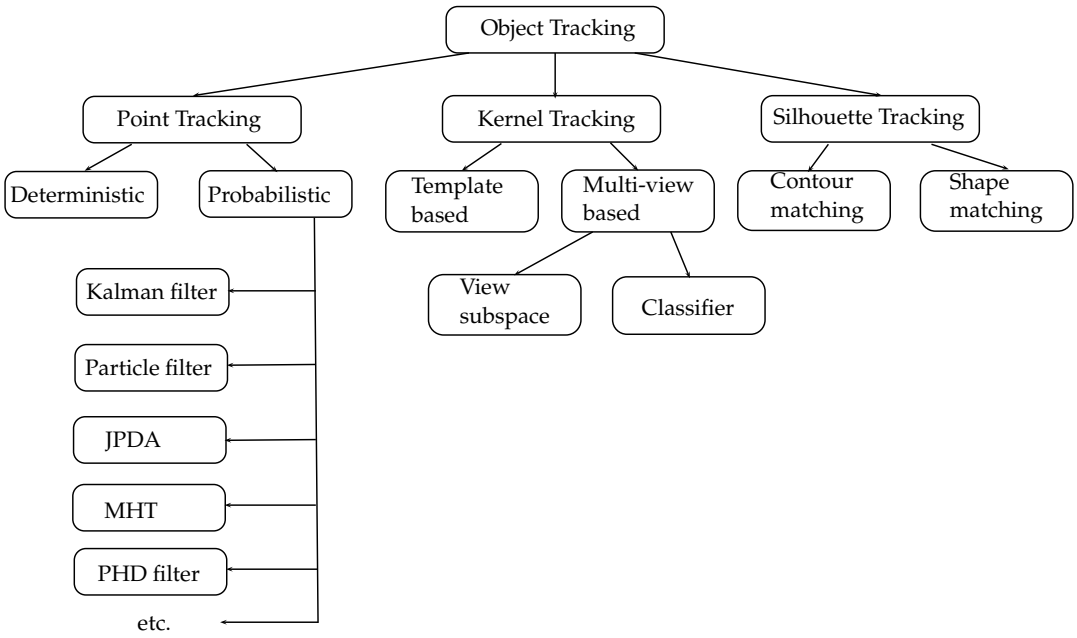


Figure 2.1 – Survey of state-of-the-art tracking methods

An example of deterministic point tracking method is presented in [Salari and Sethi 1990], where they establish correspondence for the detected points (temporal association) and extend the tracking of a missing object by adding a number of hypothetical points. Another deterministic tracking is presented by Veenman in [Veenman, Reinders, and Backer 2001], where a common motion constraint is introduced to enhance data association. This is a constraint which enforces coherent tracking of points lying on the same object.

The template-based tracking is a tracking coupled with a detection mechanism, where detectors do not look for an abstract object of given class, instead they look for a previously tracked object. For example, mean-shift tracker searches an object using histogram features and spatial positions between the track state and a new detection [Comaniciu and Meer 2002]. A layer method divides the tracking space into background level and assigns a different level for each tracked object [Tao, Sawhney, and Kumar 2002]. Each layer consists in *a priori* shapes,

motion model and a layer appearance. Layering can compensate background motion. The Kanade-Lucas-Tomasi feature tracker (KLT) [Lucas and Kanade 1981] iteratively computes the translation of region features described in [Shi and Tomasi 1994].

A classifier serving as a tracker is a Support Vector Tracker [Avidan 2001], based on a Support Vector Machines (SVM) classifier. The classifier finds an optimal separating hyperplane in the space of features for two classes of objects. In the tracking application, the classifier is trained on a difference between extracted patches of tracked region and other ones.

A multi-view based kernel tracking using view subspaces is represented by the Eigen tracking: an approach for tracking rigid and articulated objects using a view based representation [Black and Jepson 1998]. It computes the affine transformation from the current image of the object to the image reconstructed using eigenvectors. The subspace representation is build with Principal Component Analysis (PCA).

In image tracking one of the basic task is to separate tracked object from its environment, and since the most of the information in practice comes from gradients, borders, it is useful to track only a silhouette of the detected object instead of the complete area of interest. One of the shape matching approach is based on the Hough transform in the velocity space for object silhouettes [Sato and Aggarwal 2004]. The Temporal Spatio Velocity image determined there, provides a motion-based matching of the silhouettes and is less sensitive to appearance variation. Another method uses histograms of color and edges as the object models [Kang, Cohen, and Medioni 2004]. The histograms are generated from concentric circles centred on a set of control points on a reference circle, encapsulating the object silhouette.

The tracked object can also be defined in terms of spline shape and affine motion parameters. For instance, the measurements can be modelled as image edges extracted in the normal direction to the contour. The state in that case may be updated using particle filter [Isard and Blake 1998]. There are methods where the contour tracking uses temporal image gradients based on the optical flow [Bertalmio, Sapiro, and Randall 1998].

In general, the road applications do not require high precision for object shape recognition, however real-time performance is a critical feature for object tracking. Taken into account such real-time constraints, point tracking was chosen. Kernel and silhouette tracking require the use of time-cost features extraction and processing. Regarding the probabilistic methods developed in the last years, deterministic approaches have become less used.

2.2 Filtering

The term **filter** stands for methods and techniques that process sets of detections in order to reduce initial detection imprecision, to filter noise, to eliminate false positives, to cope with detection loss. In this context, a bank of filters can be employed to carry out multiple object tracking.

In this section, a short description of well-known probabilistic point tracking methods is presented. Their advantages and drawbacks are investigated. Finally, one method is then chosen for a detailed study.

2.2.1 Kalman filter

The Kalman filter is a probabilistic estimator for the linear-quadratic problem. It performs the estimates of the instantaneous state of a linear dynamic system affected by a Gaussian centred white noise [Grewal 2011; Meinhold and D.Singpurwalla 1983]. The Kalman filter is proved to be statistically optimal for any quadratic function of the estimation error in a linear system. The filter uses Bayesian inference and estimates a joint probability distribution over the variables composing filter states.

The Kalman filter model assumes a linear state evolution with a given evaluation matrix $\mathbf{F}$ of size $N \times N$, as it is shown in Eq. 2.1.

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k \tag{2.1}$$

where $\mathbf{x}_k$ of size N is an object state at a discrete time index k, $\mathbf{w}_k$ of size N represents a white noise process (each sample follows a Gaussian normal distribution with mean equals 0) with a known covariance matrix $\mathbf{Q}_k$ of size $N \times N$, which is denoted as: $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$.

The true state $\mathbf{x}_k$ is estimated based on the observation $\mathbf{z}_k$ of size M according to Eq. 2.2.

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{2.2}$$

where the matrix $\mathbf{H}_k$ of size $M \times N$ is the observation model and the vector $\mathbf{v}_k$ of length M is the known observation white noise: $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$.

Let $\hat{\mathbf{x}}_{k|k-1}$ be *a priori* state estimation at time k based on known state at time $k-1$. With this notation, $\hat{\mathbf{x}}_{k|k}$ is *a posteriori* estimation after getting observations at time k. After the definition of *a posteriori* $\mathbf{P}_{k|k}$ and *a priori* $\mathbf{P}_{k|k-1}$ error covariance matrix, the filter can be formulated through two sequential and iterative phases: prediction in Eq. 2.3 and update in Eq. 2.4.

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} && \text{Predicted (\textit{a priori}) state estimate} \\
\mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k && \text{Predicted (\textit{a priori}) estimate covariance}
\end{aligned} \tag{2.3}$$

$$\begin{aligned}
\bar{\mathbf{y}}_k &= \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} && \text{Innovation or measurement residual} \\
\mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k && \text{Innovation (or residual) covariance} \\
\mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} && \text{Optimal Kalman gain} \\
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \bar{\mathbf{y}}_k && \text{Updated (\textit{a posteriori}) state estimate} \\
\mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} && \text{Updated (\textit{a posteriori}) estimate covariance}
\end{aligned} \tag{2.4}$$

Kalman filter is limited by the linearity of state evolution and the assumption of the Gaussian nature of noise. For video and LIDAR tracking, this limitation can lead to track lost and missed association problems. Other filtering frameworks might tackle these cases.

When the tracked trajectory has a sharp turn, if the signal-to-noise ratio is kept, then the Kalman gain value is adapted with a certain latency. In that case the significance of observation

information becomes much higher than the significance of model information. The algorithm can then loose the track. Also, with a noise varying over time, the system needs to adapt the Kalman gain and covariance matrices.

2.2.2 Multiple Hypothesis Tracking

The Multiple Hypothesis Tracking presents an exhaustive method where all possible assignment combinations between track and detections are enumerated and computed in terms of probabilities [Blackman 2004; Amditis et al. 2012; C. Kim et al. 2015]. In this approach, track hypotheses are expanded to a set of new hypotheses taking into account all the possible assignments of existing tracks and the new set of measurements (i.e. observations). Each hypothesis is denoted as an object-to-track assignment.

An illustration of a hypothesis tree is presented in Fig. 2.2.

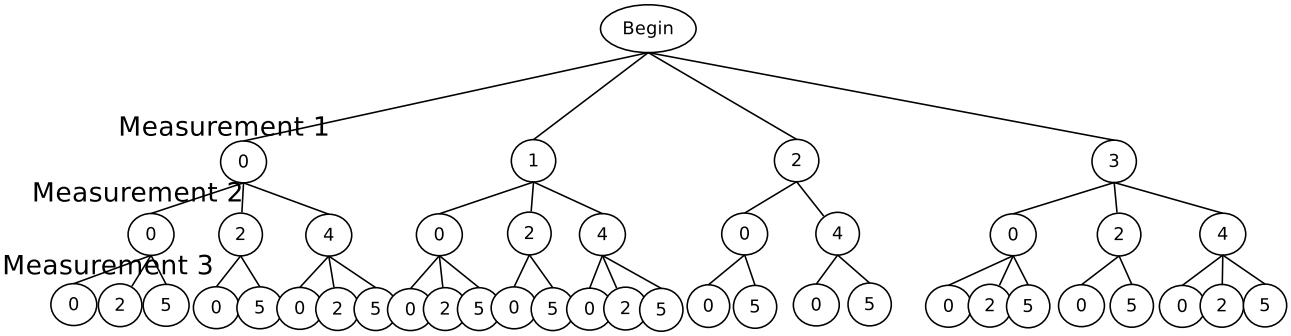


Figure 2.2 – Multiple hypothesis scheme. Node numbers indicate to which track the measurement is assigned to. "Zero" stands for "false alarm" assignment, other numbers are existing or new tracks. New measurements generally increase the number of hypotheses

For each hypothesis Ω_i^k, formed from the parent hypotheses Ω_g^{k-1}, its probability P_i^k is calculated according to Eq. 2.5 from probability P_g^{k-1} and other values. Here k is a measurement number, representing also the MHT tree level as in Fig. 2.2 where i, g are respectively hypotheses indices at levels k and $k-1$.

$$P_i^k = \frac{1}{c} P_D^{N_{DT}} (1 - P_D)^{(N_{TGT} - N_{DT})} \beta_{FT}^{N_{FT}} \beta_{NT}^{N_{NT}} \times \Big[\prod_{m=1}^{N_{DT}} N(\mathbf{z}_m - \mathbf{H}\mathbf{x}_g, \mathbf{B}) \Big] P_g^{k-1} \tag{2.5}$$

where:

— P_D, Probability of detection
— β_{FT}, Density of false targets
— β_{NT}, Density new detected targets detected
— N_{DT}, Number of measurements associated to prior targets
— N_{TGT}, Number of all known prior targets
— N_{FT}, Number of measurements associated to false targets
— N_{NT}, Number of measurements associated to new targets
— c, Normalization constant
— $\mathbf{x}_g$ is a track supported by the hypothesis Ω_g^{k-1}

— $\mathbf{z}_m - \mathbf{H}\mathbf{x}_g$ and $\mathbf{B}$ are the innovation vector and innovation covariance matrix respectively.

Evaluating the probabilities of hypotheses after the arrival of new observations, and by using an additional mechanism of rejecting improbable hypotheses, one can track one or more objects by assigning them to the hypotheses tree.

The drawbacks of MHT are:

— Exponential increasing of hypotheses number. The exponential complexity can be limited by using hypotheses pruning strategies, but even then, the computational cost remains high.
— Multiple hypothesis tracking method in its initial form provides only temporal association between detections without any prediction.
— MHT has no possibility to filter or correct noisy detections.

The two last disadvantages can be managed when MHT is completed by a single-object tracker, like Kalman filter. This combination is called Multiple Hypothesis Kalman filter (MHKF) [Bazzani, Bloisi, and Murino 2009].

2.2.3 Joint Probabilistic Data Association Filter (JPDAF)

Joint Probabilistic Data Association (JPDA) is an object-to-track association method based on a joint probability estimation [Bar-Shalom and Tse 1975; Rezatofighi et al. 2015; I. J. Cox 1992].

Let $\mathbf{x}_k^1, \ldots, \mathbf{x}_k^N$ and $\mathbf{z}_k^1, \ldots, \mathbf{z}_k^M$ be the states of N tracks and M detections at moment k respectively. The assignment probability representing that the measurement i is generated by the target j, is denoted as $p_k(d_i^j)$:

$$p_k(d_i^j) \propto \begin{cases} (1 - p_D)\beta & \text{if i=0} \\ p_D \cdot \mathcal{N}(\mathbf{z}_k^i; \hat{\mathbf{x}}_k^j, \mathbf{S}_S) & \text{otherwise} \end{cases} \tag{2.6}$$

where $\hat{x}_k^j$ is the predicted state of the object j at time k, p_D is the detection probability, β is a false detection density, and $\mathbf{S}_S$ is the innovation covariance matrix of the Kalman filter. The JPDA calculates joint probabilities $q_k(d_i^j) = q_k(d_i^j = 1)$ on the joint data association space Θ, which contains all possible combination pairs between detections and tracks, satisfying the following constraints:

$$\begin{aligned} \Theta = \Big\{ \theta = (d_i^j)_{i \in [M]_0, j \in [N]} \Big\| \, & d_i^j \in \{0, 1\} \\ \wedge \quad & \sum_{j=1}^{N} d_i^j \leq 1, \quad \forall i \in [M] \\ \wedge \quad & \sum_{i=0}^{M} d_i^j = 1, \quad \forall j \in [N] \Big\} \end{aligned} \tag{2.7}$$

where $\theta \in \Theta$ is a binary vector denoting one solution of the data association problem. For a subset θ_i^j including hypotheses which assign detections i to target j, the JPDA probability

$q_k(d_i^j)$ is calculated as:

$$\begin{aligned} q_k(d_i^j) &= \sum_{\theta \in \theta_i^j} p(\theta) \\ p(\theta) &= \prod_{\substack{\forall m \in [M]_0 \\ \forall n \in [N]}} (p_k(d_m^n))^{d_m^n} \end{aligned} \tag{2.8}$$

The solution maximizes the joint probability of Eq. 2.8.

As MHT, JPDA does not provide filtering of noisy detection, neither does allow for prediction. Analogously to MHKF, the JPDA can be used in a combination with Kalman filtering. The resulted method is called JPDAF. An advantage of JPDA with respect to MHT is its computationally cost.

2.2.4 Particle filter

The Bayesian Recursive Filtering is a filtering method based on the construction of posterior probability density function $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, where $\mathbf{x}_k$ is a state vector of size N at moment k and $\mathbf{z}_{1:k}$ are all observation vectors of size M from moment 1 to moment k. In this method, the posterior probability density is recursively calculated with two processing steps: prediction and update.

Prediction According to the Bayes' rule, using Chapman-Kolmogorov integral form, one can get an expression for the prediction step:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int_{\mathcal{X}} p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \tag{2.9}$$

where $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the prior probability of state $\mathbf{x}_k$ based on known previous state $\mathbf{x}_{k-1}$, $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ is the posterior probability of state $\mathbf{x}_{k-1}$ based on known observations $\mathbf{z}_{1:k-1}$, $\mathcal{X}$ is a state space. In discrete form, Eq. 2.9 becomes:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \sum_{\mathbf{x}_{k-1} \in \mathcal{X}} p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \tag{2.10}$$

Update Using new observations at time k, the prior probability density function can be updated as follows:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int_{\mathcal{X}} p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) d\mathbf{x}_k} \tag{2.11}$$

or

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\sum_{\mathbf{x}_k \in \mathcal{X}} p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1})} \tag{2.12}$$

The Particle filter represents the posterior probability density function as a discrete number of samples, called particles [Ng and Delp 2009; Arulampalam et al. 2002; Hermes et al. 2009]. A particle represents a hypothesis of the states and it is randomly placed around *a prior* density. After sampling, particles are propagated according to the evolution model, and weights are assigned to them according to a likelihood model. A particle's weight is proportional to its

likelihood: particles located close to new observations have high weights, and particles far away from new observations have low weights.

The object state can be reconstructed as a weighted mean of particle states, evolving according to the state model. Its probability density is defined as:

$$p(\mathbf{x}_k) \approx \frac{1}{K} \sum_{i=1}^{K} \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{2.13}$$

where δ is a Kronecker delta function, $\mathbf{x}_i$ is a state of i^{th} particle and K is the number of particles

Particle's weights are measured by means of a likelihood model $\mathcal{L}$, and the posterior probability density function $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be approximated as follows:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{1:k}) &= \sum_{i=1}^{K} \omega_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \\ \omega_k^i &= \frac{\mathcal{L}(\mathbf{z}_k|\mathbf{x}_k^i) p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}, \mathbf{z}_k)} \\ \sum_{i=1}^{K} \omega_k^i &= 1 \end{aligned} \tag{2.14}$$

where $q(x_k^i|x_{k-1}, z_k)$ is the normalization term.

The particle filter does not require linearity of the evolution model and the noise is not assumed to be Gaussian, as for the Kalman filter.

The particle filter model has a precision depending on the number of using particles: more particles provide more stability for the stochastic process. In the initial form, the particle filter is not adapted to multi-object tracking. However, there is a set of algorithms using the particle filter principle. One of them, PHD filter, is shown further.

2.2.5 Probability Hypothesis Density (PHD) filter

The Probability Hypothesis Density filter is a multi-object tracker based on the propagation of probability hypothesis densities, as the first order moment of the multi-object posterior. The PHD can be analysed so as to observe the number of tracked objects [Y. Zheng et al. 2013]. It has various implementations, including Sequential Monte Carlo (SMCPHD) [Vo, S. Singh, and Doucet 2003], Gaussian Mixture [Vo and W.-K. Ma 2006], Box-PHD implementation [Schikora et al. 2014], etc. The PHD can be represented in the form of particles, and in that case the PHD filter remains a particle filter, but with some new multi-object features.

The PHD filter is efficient in terms of computational resources [Maggio, Piccardo, et al. 2007]. A PHD is capable of filtering clutter, completing missing observations and coping with noisy ones.

Recently, efficient particle implementations of PHD filtering have been proposed [Vo, S. Singh, and Doucet 2003; Maggio, Piccardo, et al. 2007; Schikora et al. 2014]. These approaches include the estimation of the number of observed tracks. To this end, particle clustering is necessary to identify tracks. However, such a procedure is non-trivial in urban scenarios where

objects move in close to each other.

PHD is based on the propagation of first-order moment of the multi-object posterior. This implies that the posterior is a condensation of random finite set (RFS). Based on this last statement and some other notions, a generic particle implementation of the PHD filter algorithm can be formulated structured by prediction, update and resampling steps, detailed below.

Let define some notation:

— k, Time index, filter iteration
— L_k, Number of particles at k^{th} iteration
— J_k, Number of new born particles
— i, Particle index
— Z_k, Set of all observations at k^{th} iteration
— ξ_k^i, i^{th} particle at k^{th} iteration
— w_k^i, Weight, attributed to the particle ξ_k^i
— $\hat{\xi}_k, \hat{w}_{k|k-1}$, Prior particle state and prior weight estimation
— $\phi(\xi_i, \xi_j)$, Prediction operator
— $\psi(\xi_i, \xi_j)$, Update operator
— $\gamma(\xi)$, PHD of spontaneously appeared objects, i.e. of new particle ξ
— $v(\xi)$, Probability of non-detection
— $q_k(\xi_i|\xi_j, Z_k)$, Probability of sampling ξ_i from ξ_j, taking into account observations Z_k
— $p_k(\xi_i|Z_k)$, Probability of sampling new particle ξ_i
— κ_k, Clutter probability density

Prediction

For i=1, ..., L_{k-1}, sample $\hat{\xi}_k^i \sim q_k(\cdot|\xi_{k-1}^i, Z_k)$ and compute the predicted weights:

$$\hat{w}_{k|k-1}^i = \frac{\phi(\hat{\xi}_k^i, \xi_{k-1}^i)}{q_k(\hat{\xi}_k^i|\xi_{k-1}^i, Z_k)} w_{k-1}^i \tag{2.15}$$

For i=$L_{k-1}+1, \ldots, L_{k-1}+J_k$, sample $\hat{\xi}_k^i \sim p_k(\cdot|Z_k$ and compute the weights of new born particles:

$$\hat{w}_{k|k-1}^i = \frac{1}{J_k} \frac{\gamma(\hat{\xi}_k^i)}{p_k(\hat{\xi}_k^i|Z_k)} \tag{2.16}$$

Update

For each $z \in Z_k$, compute:

$$C_k(z) = \sum_{j=1}^{L_{k-1}+J_k} \psi_{k,z}(\hat{\xi}_k^j) \hat{w}_{k|k-1}^j \tag{2.17}$$

For i=1, ..., $L_{k-1}+J_k$ update weights

$$\hat{w}_k^i = \Big[v(\hat{\xi}_k^i) + \sum_{z \in Z_k} \frac{\psi_{k,z}(\hat{\xi}_k^i)}{\kappa_k(z) + C_k(z)}\Big] \hat{w}_{k|k-1}^i \tag{2.18}$$

Resampling

Compute the total mass $M_{k|k} = \sum_{j=1}^{L_{k-1}+J_k} \hat{w}_k^j$

Resample $\left\{\hat{w}_k^i/M_{k|k}, \hat{\xi}_k^i\right\}_{i=1}^{L_{k-1}+J_k}$ to get $\left\{w_k^i/M_{k|k}, \xi_k^i\right\}_{i=1}^{L_k}$

This implementation does not allow for tracks' temporal association, i.e. the number of tracks for each rayon of interest can be statistically estimated, but cannot reconstruct their trajectories. In next section, the detailed implementation of a tracker based on particle PHD is presented.

2.3 Proposed multi-object tracking system

In the variant of the particle-based PHD-filter presented below, the problems of clustering and cardinality estimation are avoided by initializing tracks with a fixed number of particles constantly associated to them. The proposed method is not claiming to perform superior multi-object tracking, however, it does facilitate the integration of contextual information as is is shown in Chap. 4, because particles facilitates context data representation.

In this work a slight modification of the SMCPHD filter is used, with an integrated data association mechanism, used to remove old, useless targets and to create new ones. The follow detailed explanation concerns only this particular implementation of the filter, and can not be considered as generic.

2.3.1 Filter implementation

The PHD filter is represented by N^x dynamically changing tracks $\mathbf{x}_m$, $m \in 1..N^x$. Each track $\mathbf{x}_m$ contains N^p particles. Particle $\xi_{\mathbf{x}_m,n}$, $n \in 1..N^p$ contains a set of vectors $\{[c_i, d_i, v_i]^T\}$, $i \in 1..D$, where D is a space dimension, c_i is the center coordinate, d_i is the detection size and v_i is the speed. Here index i is common for all three characteristics of a state (or a particle). That means all three vectors have the same dimensionality. A weight $\omega_{\mathbf{x}_m,n}$, $n \in 1..N^p$ is assigned to each particle $\xi_{\mathbf{x}_m,n}$, $n \in 1..N^p$.

The size and velocity were introduced in the model to efficiently resolve the cases of tracks intersections. At the moment of tracks occlusion, the position states become equal. With the presence of the speed and size, the distinction between tracks is kept during some time before the adaptation.

The current implementation is a one-to-one approach (i.e. mono-hypothesis), that is a track can represent only one object and one object can not have more than one associated track.

The parameters of the PHD filter are:

— R_d, death rate
— R_b, birth rate
— R_{ret}, retard rate

The tracking process implements the following stages: prediction, association, observation, resampling, merging and correction.

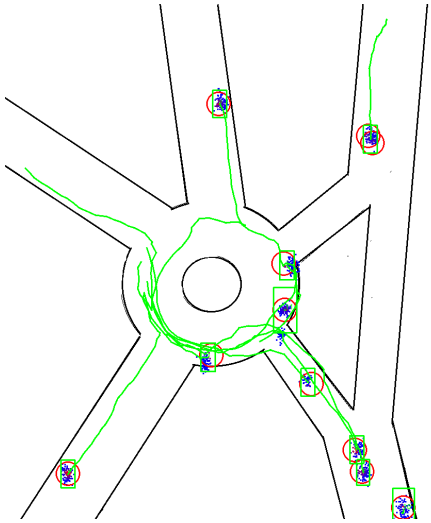


Figure 2.3 – Example of a simulated multi-target tracking scene. Green rectangles are tracks, red circles are detections, dots are particles

Prediction Tracks and theirs particles are propagated according to the motion model:

$$\begin{aligned} c_{i,k|k-1} &= c_{i,k-1} + v_{i,k-1} \\ d_{i,k|k-1} &= d_{i,k-1}, \qquad\qquad i \in 1..D \\ v_{i,k|k-1} &= v_{i,k-1} \end{aligned} \tag{2.19}$$

where k is a discrete time moment and $k-1$ is a previous time moment. The position coordinates are evaluated linearly according to the current speed. The object size and speed change slowly and their adaptations can be handled by only particles resampling.

Association Input observations $\mathbf{z}_j$, where $j \in 1..N^z$ and N^z is a number of observations, are assigned to existing tracks $\mathbf{x}_m$, $m \in 1..N^x$. Tracks assigned to observations increase their life rate by R_d:

$$R_{m,k} = \min(R_{m,k-1} + R_d, 1), \quad m \in 1..N^x \tag{2.20}$$

where the life rate $R_{m,k}$ for a track x_m is a value defining if this track must be eliminated as out of date or be still evaluated. Non-associated tracks update their probability by:

$$R_{m,k} = \max(R_{m,k-1} - R_d, 0) \tag{2.21}$$

Observations which are not associated give birth to new tracks with an initial life rate equal to the birth rate:

$$R_{m,k} = R_b \tag{2.22}$$

In case when a new track is created, the weight of its particles are:

$$\omega_{\mathbf{x}_m,n} = \frac{R_{m,k}}{N^p}, \quad n \in 1..N^p \tag{2.23}$$

The association algorithm represents an implementation of Global Nearest Neighbor (GNN) algorithm. It was chosen because it is simple in implementation, more precise in tracking applications than Suboptimal Nearest Neighbour (SNN) [Konstantinova, Udvarev, and Semerdjiev

2003].

In detail, the association algorithm's steps are:

1. For all pairs $\{\mathbf{z}_j, \mathbf{x}_m\}$, $j \in 1..N^z; m \in 1..N^x$ the pseudo-distance $G(\mathbf{x}_{m,k}, \mathbf{z}_{j,k})$ is calculated, where the function $G()$ is a product of Gaussians [Sattarov, Sergio Alberto Rodríguez Florez, et al. 2014]:

$$\begin{aligned} G(\mathbf{x}_{m,k}, \mathbf{z}_{j,k}) = \mathcal{N}(\|\mathbf{c}_{\mathbf{x}_m} - \mathbf{c}_{\mathbf{z}_j}\| \quad |0, K_c) \\ \times \mathcal{N}(\|\mathbf{d}_{\mathbf{x}_m} - \mathbf{d}_{\mathbf{z}_j}\| \quad |0, K_d) \\ \times \mathcal{N}(\|\mathbf{v}_{\mathbf{x}_m} - \mathbf{v}_{\mathbf{z}_j}\| \quad |0, K_v) \end{aligned} \tag{2.24}$$

   where K_c, K_d, K_v are coefficients in the range $]0, 1]$, are experimentally chosen, $\mathbf{c}, \mathbf{d}, \mathbf{v}$ are vectors of center coordinates, detection size and speed.

2. Find the nearest pair following the equation:

$$(\mathbf{x}, \mathbf{z}) = \underset{\mathbf{x}_m \in X, \mathbf{z}_j \in Z}{\operatorname{argmax}} G(\mathbf{x}_m, \mathbf{z}_j) \tag{2.25}$$

   Associate $\mathbf{x}$ and $\mathbf{z}$, remove $\mathbf{x}$ from list of pairs to associate and repeat this step if $G(\mathbf{x}, \mathbf{z}) > \theta^a$, where θ^a is an empirically chosen threshold.

3. Finally, a list of associated pairs $\{\mathbf{z}_j, \mathbf{x}_m\}$, a list of non-associated detections $\{\mathbf{z}_j\}$ and a list of non-associated tracks $\{\mathbf{x}_m\}$ are obtained.

Observation For each new observation $\mathbf{z}_{j,k}$, $j \in 1..N^z$, and for each particle $\xi_{\mathbf{x}_m,n}$, $\mathbf{x}_m \in X$, $m \in 1..N^x$, $n \in 1..N^p$ a pseudo-distance is calculated: $G(\xi_{\mathbf{x}_m,n}, \mathbf{z}_{j,k})$. The pseudo-distances are normalized relative to observations:

$$\omega_{\mathbf{x}_m,n} = (1 - R_{ret}) \sum_{j=1}^{N^z} \frac{G(\xi_{\mathbf{x}_m,n}, \mathbf{z}_j)}{\sum_{l=1}^{N^p} \sum_{s=1}^{N^x} G(\xi_{\mathbf{x}_s,l}, \mathbf{z}_j)} + Previous(\omega_{\mathbf{x}_m,n}) \times R_{ret} \tag{2.26}$$

The term $\omega_{\mathbf{x}_k,n} \times R_{ret}$ represents the "old" particle weights. It is used to add an inertia component to a track, to reduce useless fluctuations. The normalization term has a summarizing on distances to all tracks. That came from PHD filter principle, where particle represent first-order moment. That means that one observation must bring weights equivalent to

Resampling Track $\mathbf{x}_m$ is deleted if its current life rate $R_{m,k} < \theta^d$. Here θ^d is an arbitrary parameter. These threshold shows that if during some time no one detection was associated to these track (i.e. its life rate decreases) it is considered as lost and must be deleted. Otherwise, if $R_{m,k} >= \theta^d$ its particles are randomly sampled using:

$$\begin{aligned} c_{i,k} &= c_{i,k|k-1} + \mathcal{N}_c(0, \hat{K}_c) \\ d_{i,k} &= d_{i,k|k-1} + \mathcal{N}_d(0, \hat{K}_d) \\ v_{i,k} &= v_{i,k|k-1} + \mathcal{N}_v(0, \hat{K}_v) \end{aligned} \tag{2.27}$$

Here $\mathcal{N}_c, \mathcal{N}_d, \mathcal{N}_v$ are different white noises, and $\hat{K}_c, \hat{K}_d, \hat{K}_v$ are coefficients, chosen empirically.

The parameters $\hat{K}_c, \hat{K}_d, \hat{K}_v$ are proportional to the value $\frac{1}{R_{m,k}}$. This is done in order to make particles more dispersed when a track is "lost". The increased radius of particle distribution increases the chances that some of particles will be near detections.

Merging If two tracks $\mathbf{x}_{m1}$ and $\mathbf{x}_{m2}$ have a pseudo-distance $G(\mathbf{x}_{m1}, \mathbf{x}_{m2}) > \theta^m$, they are supposed to belong to a single object, and the newer track is deleted. Here θ^m is a predefined merging threshold.

There are different situations when a merging mechanism is required:

1. One object was detected several times and those detections were recognized as belonging to different objects. During the association step, a new track was created near the old one. After some time they approach and a merging mechanism removes the newer track. This is an ideal case explaining why the merging step was considered. Without it, a set of tracks following the same object would be created. That contradicts mono-hypothesis (one-to-one) tracking defined above. The newer track hypothesis is pruned since it encloses less information than the older one.
2. Two different objects get into an occlusion scenario: one track overlaps another track. The merging step deletes the newer one and when the occlusion is finished, a new track will be created. This scenario is undesirable, but the track lost is partially compensated with a new track creation.
3. Two object get into occlusion and one track quits the tracking space while been occluded. Here there is no problem: the track is correctly pruned in the merging procedure.

Correction New track centres are computed as the mean of particles associated to that track:

$$\mathbf{x}_{m,k} = \frac{1}{N^p} \sum_{n=1}^{N^p} \xi_{\mathbf{x}_m,n,k} \tag{2.28}$$

The resulted mean is weighted because after the resampling step all particles are equiprobable. The output of the algorithm is represented as a set of tracks.

This tracking system is used in recent publications: [Gepperth, Ortiz, et al. 2016; Gepperth, Sattarov, et al. 2014].

2.3.2 Evaluation method

The approach was tested on simulated data and on the public KITTI benchmark dataset [Fritsch, Kuehnl, and Geiger 2013] using annotated tracklets as Ground Truth. The common schema to evaluate results requires four sets of data:

1. The set of labeled rectangles representing tracks constructed by tracking algorithms: X.
2. The set of labeled rectangles representing real objects Y, or Ground Truth.
3. The set of labeled rectangles representing noisy objects Z. It is obtained from Ground Truth by artificially introducing missed (false negative) detections, and by corrupting retained detections by noise. Noise is modelled as an additive Gaussian fluctuation

applied to positions and sizes (c_i, d_i), $i \in 1..D$ of all Ground Truth objects. Each noisy detection $\mathbf{z} \in Z$ has a Ground Truth pair $\mathbf{y} \in Y$.

4. The set of pairs of labels representing associations between noisy detections and tracks noted as XZ^{assoc}

As the particle implementation of PHD-filtering contains a pseudo-random process, small variations can occur over trials. To precisely calculate the evaluation criteria, the results are calculated as the mean and the standard variation of both measurements across 20 trials.

Two major evaluation criteria are used to quantify the accuracy of the approach: an overlap criterion and a continuity criterion. The overlap criterion measures the accuracy of a track's position with respect to associated real object position. The "continuity" criterion computes the quality of object-to-track associations.

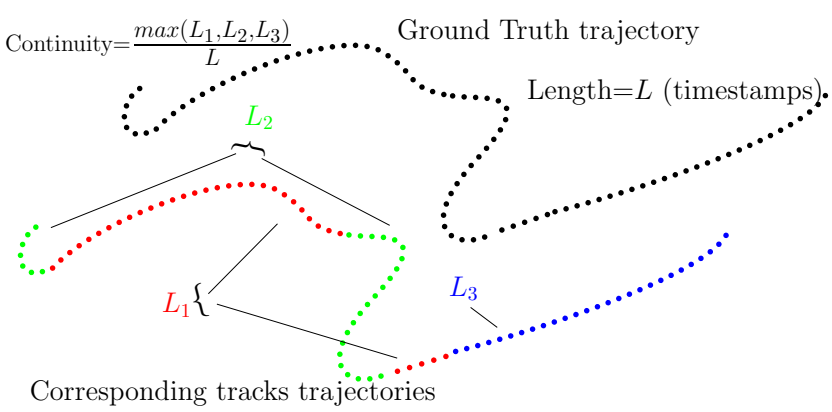
Track overlap criterion is derived from [Yin, Makris, and Velastin 2007; Manohar et al. 2006] as the mean of all association overlaps (i.e. overlap between track and real object):

$$Overlap = \frac{1}{N^{assoc}} \sum_{(k,i)} \max\left(\frac{S(x_k \cap y_i)}{S(x_k)}, \frac{S(x_k \cap y_i)}{S(y_i)}\right) \tag{2.29}$$

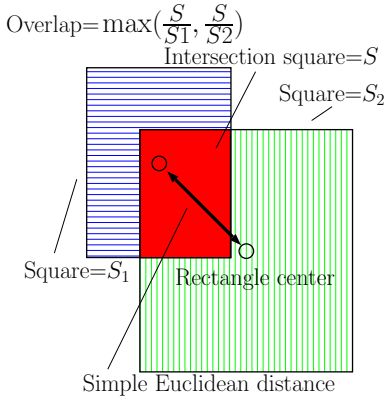
where N^{assoc} is a number of associations in XZ^{assoc} and S$(\cdot)$ is an area occupied by detection. This form was chosen because it:

1. Takes the detection size into account.
2. Has a normalized values in $[0, 1]$ interval
3. Penalizes not only the distance between a detection and a track state, but the incorrect size of the track state, both for too large and too small.
4. Is integral for all tested scenario.

The overlap value is always $\in [0, 1]$, where 1 represents the ideal case of full overlap. The illustration is provided on Fig. 2.4b.



(a) Continuity measure illustration. One must note that the disconnected parts of the same track are calculated together



(b) Overlap measure with simple Euclidean distance illustration

Figure 2.4 – Tracking evaluation criteria

The "continuity" criterion is derived from [Smith et al. 2005; Holt et al. 2010] and is calculated according to the formula:

$$Continuity = \frac{1}{N^Y} \sum_{y_i \in Y} \max_k \frac{1}{N^{y_i}} \sum_t \delta_{k,i}(t) \tag{2.30}$$

where N^Y is a number of Ground Truth objects, N^{y_i} is the number of appearances of the object y_i during the whole tracking scenario, $\delta_{k,i}(t) = 1$ if $(k,i) \in XZ^{assoc}(t)$ and $\delta_{k,i}(t) = 0$ otherwise. The continuity measure thus describes the mean of the longest associations. It varies in $]0,1]$, where 1 is the ideal case of constant associations. The illustration is provided on Fig. 2.4a. The proposed continuity formula is used because it:

1. Penalizes changes of tracks associated with GT states.
2. Distinguishes the short-time and long-time changes in GT-tracks association.
3. Processes together two long GT-track association separated in time, but identical in the labels.

2.3.3 Tests

Before testing the tracking system on a new approach, it is useful to test the proposed tracking itself. To this end, the evaluation criteria described above are used on a KITTI dataset.

The tracking tests are effectuated in 2D space of detections, where points are represented by their 2D position, size and velocity. Two types of tracking spaces are used:

— Camera image projection, where the actual "detection size" depends on the distance between camera and the observed object. The position, size and velocity are measured in pixels.

— The Bird-eye-view ground plane. The size of detections are supposed constant for each object and represent a square with side equals $max(width, length)$, where $width$ and $length$ are given real object sizes obtained from KITTI dataset. The position, size and velocity are measured in metres.

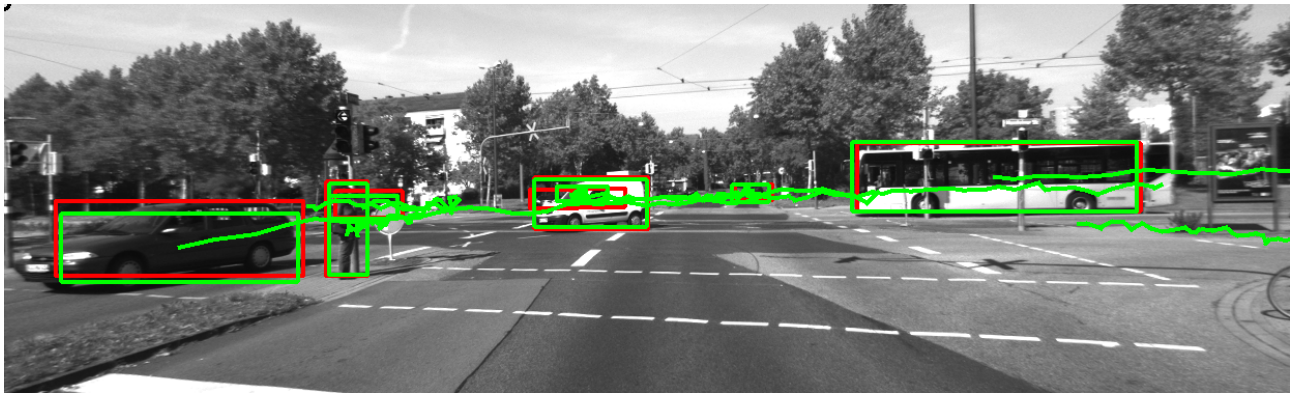
The visual tracking quality estimation can be observed at Fig. 2.5a and Fig. 2.5b. It is obvious that the visual tracking is a more complicated task, as the track states can change their sizes, can create occlusions and move too fast when approaching to the camera.

Now, the tracking quality can be evaluated with defined criteria "continuity", "overlap" and simple Euclidean distance. Two KITTI scenarios are tested both for camera and bird-eye views in two modes:

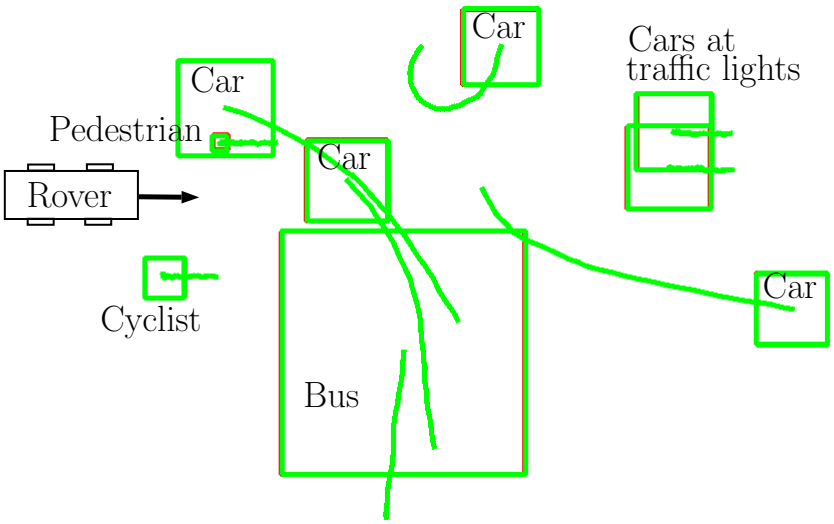
— Full KITTI tracklets are provided as detections. In this case the detections are tight in time and the tracking is supposed to be easy.

— Noised scenario: only one third (two thirds for camera view) of KITTI reference positions are served as detections. They are randomly chosen. The absent detections imitate the false negatives. Thus tracking process is supposed to be more complicated.

Results are shown in Fig. 2.6 for bird eye view and in Fig. 2.7 for camera view. The tracking in camera view is far from being perfect, but in bird eye view without false negatives the continuity of tracks are 100% respected. It is worth noting that the matching criteria (distance and overlap) on a noised scenario provides a higher rate than for the noise-free scenario. Moreover,



(a) Camera-view



(b) Bird-eye-view

Figure 2.5 – Visual illustration of tracking system. Red rectangles are detections, green rectangles are actual tracks state. The green lines are tracks recent trajectories. One can see that camera-view tracking is a more difficult task than in bird-eye-view. The scene is the same in both figures

a decreasing on continuity criterion are correlated to track-object association errors. However, a track-object association error can induce new track creation improving distance and overlap criteria. As camera view tracking is not stable enough, bird eye view will be used to test the approach proposed in Chap. 4.

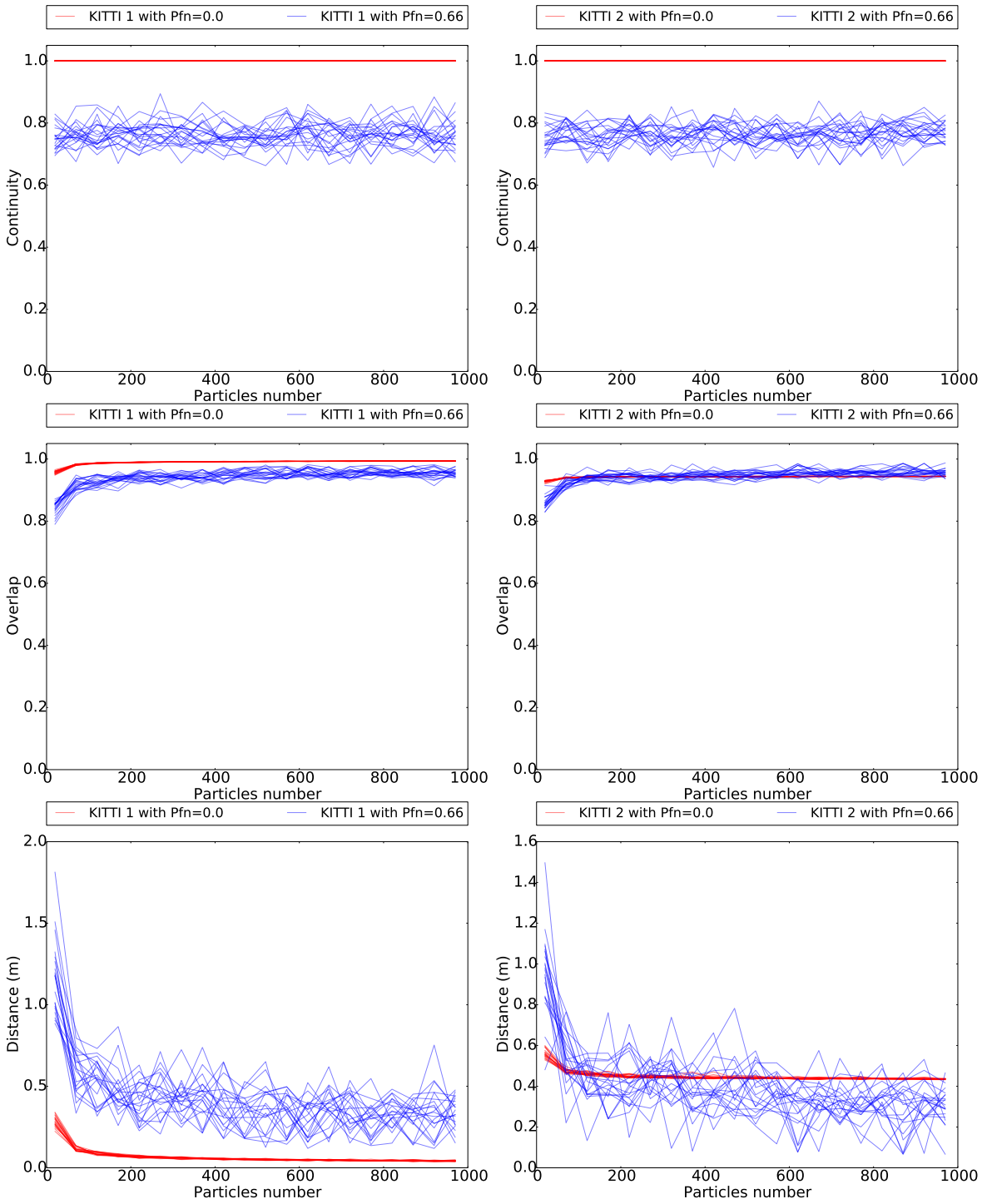


Figure 2.6 – Continuity, overlap and Euclidean distance for tracking in bird eye view. Two KITTI scenarios are the sources of detections: one is on the left column, other - on the right column. As the particle implementation brings a random component, for each configuration 20 essays was repeated to have a representative statistics. Two cases was tested: all detections provided and 33% of detections provided.

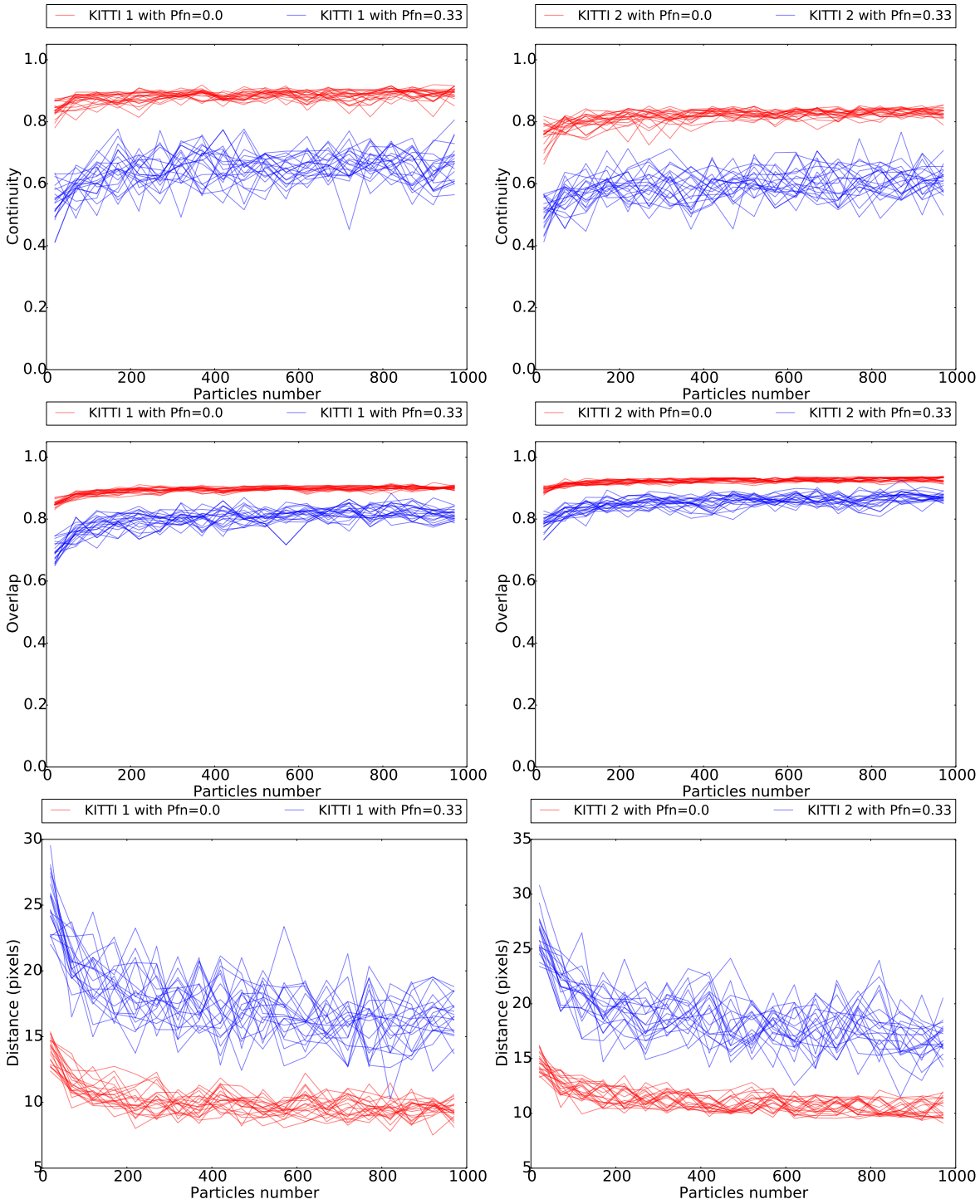


Figure 2.7 – Continuity, overlap and Euclidean distance for tracking in image projection view. Two KITTI scenarios are the sources of detections: one is on the left column, other - on the right column. 20 essays was repeated to have a representative statistics. Two cases was tested: all detections provided and 66% of detections provided.

Chapter 3

Multi-sensor data association

Contents

3.1 Introduction

While calibration approaches are often quite precise, the calibration procedure itself is complex and error-prone and requires considerable expertise. Furthermore, a calibration procedure depends intrinsically on the common data representation (e.g. calibration pattern, features), and needs to be re-designed every time a change is made. On the other hand, it is often rather easy and cheap to obtain a large number of sample measurements from both sensors. Assuming the existence of such a sample dataset, a method to extract an implicit calibration model between vision and LIDAR sensors is proposed. A data-driven approach where the statistics of each sensor are used to optimally project both measurements (i.e. object-level) onto a standardized representation format is pursued to be able to apply generic probabilistic methods.

In this way, this approach is completely independent of the intrinsic characteristics of the measurements, and in particular of their dimensionality (i.e. n-d observations), leading to a strong reduction of design and re-design effort for the conception of multi-modal processing system in vehicles.

A learning approach that allows to detect correspondences between multiple sensors measurements is presented. In contrast to approaches that rely on calibration (see Sec. 1.3.3), a learning approach creates an implicit calibration model from training data.

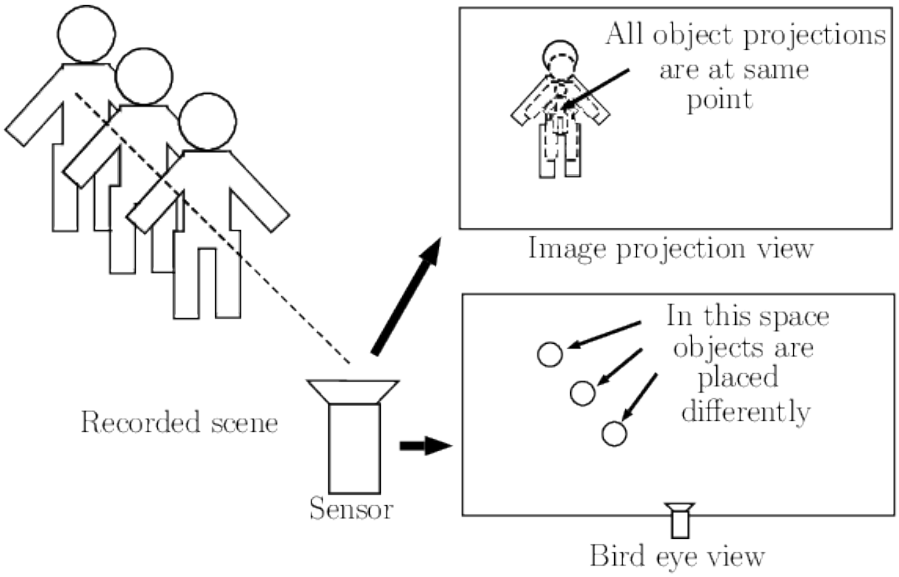


Figure 3.1 – Illustration of the case, when the unique transform between sensor's spaces does not exist: when detected by some sensors, an object can be viewed as one detection, for other sensors the same object can be viewed as a distribution of probable detection

The model can provide three functions: first of all, it converts a measurement from one sensor into the coordinate system of an other sensor, or into a distribution of probable measurements in case where a transformation is not unique, like it is illustrated in Fig. 3.1. Secondly, the model is able to decide if two visual/LIDAR measurements are likely to come from the same object. This is of profound importance for applications such as object detection or tracking where contributions from several sensors need to be combined [Hall, Liggins, and Llinas 2009].

The feasibility of the approach is demonstrated by training and evaluating the system on tracklets in a KITTI dataset as well as on a small set of real-world scenes containing pedestrians, in which the method finds correspondences between the results of raw camera and LIDAR-based detections [Sattarov, Gepperth, et al. 2015].

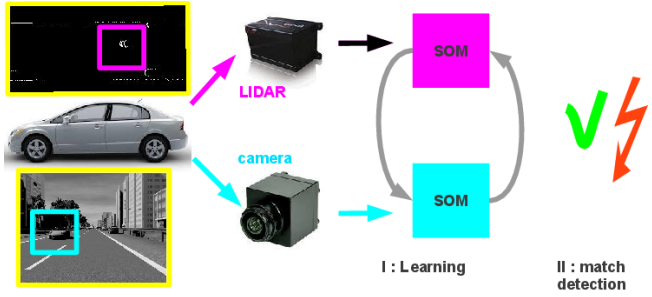


Figure 3.2 – Block architecture of the proposed correspondence detection method.

A new way of detecting multi-modal correspondences for the important vision/LIDAR sensor combination is presented here. A main contribution of the used learning approach is that the "calibration" procedure is much simpler, it uses Self-Organizing Maps (SOM) detailed in Sec. 3.3.2, and so it can in fact be handled by a non-expert regardless of the precise type of measurements that are conducted. Furthermore, it is shown that the resulting data alignment is very computationally efficient and sufficiently accurate for most applications. Performing all experiments using the publicly available dataset adds significant credibility to the results.

The complete model is composed of several components, as visualized in Fig. 3.2:

— LIDAR and vision sensor
— Means to measure interesting quantities in both common reference data
— Self-organized Maps (SOM) for vision and for LIDAR, which learn to represent the inputs coming from the respective (synchronous) measurements
— An algorithm for learning a correspondence model between SOMs
— A module for deciding when two measurements correspond, based on the SOMs and the learned correspondence model

Not to complicate the clean and simple algorithm, only unimodal processing in each modality are proposed by details

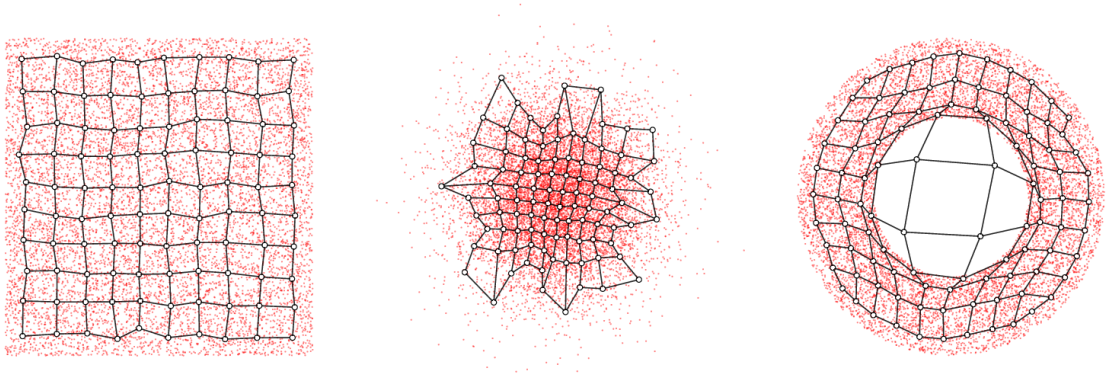
3.2 Self-Organizing Maps (SOM)

Self-Organizing Map (SOM), also called Kohonen map [Kohonen 1982], is an artificial neural network providing unsupervised learning. As a result of competitive learning a SOM provides a low-dimensional (the most frequently - 2D) map, which is a discrete representation of the training samples space. SOMs are useful for high-dimensional data visualisation and clustering. 2D SOM represents a grid of nodes. While training, the training samples drag the nearest node and its grid neighbours to the samples. As a result, the nodes repeat the training data spatial structure.

SOMs are intuitively simple and their various applications in classification and clustering prove the SOMs efficiency. As disadvantages one can mention the poor generalization at extremes of training data. Also the data representation surface must be sufficiently continuous, otherwise some part of nodes will be placed in area without samples.

At Fig. 3.3 some trivial examples of 2D data covering by SOMs are shown.

In autonomous vehicle and tracking domains, the SOMs are applied for vehicle future movement direction. In [Bohlooli and Jamshidi 2012] the repetitive car's trips composed of movement patters are learned and reproduced by SOMs. In [J. Cho et al. 2006] SOMs learn to choose unmanned motion models using sensor's signals for embedded control of an autonomous robot. SOMs are used to classify images of frontal camera [Neagoe and Tudoran 2008]. Each image there can be classified as "sharp turning left", "sharp turning right", "smooth turning left, smooth turning right", "moving forward". In [Hendzel 2005] the SOMs are trained to find a collision free path for an autonomous robot. The SOM response consists in a motion model coefficients, affecting the robot behaviour. The node weights corrections are effectuated based on robot's observations online.



(a) (b) (c)

Figure 3.3 – Examples of SOMs functionality on 2D sample data: a) Samples are uniformly distributed b) Normally distributed c) Represent a ring. Red dots are training samples, white circles are SOM nodes, black lines connected nodes are the inner SOM neighbours relations. All SOMs are grids of 10×10 nodes

For tracking purposes the feature-based recognition of lost tracks is realised in [Bevilacqua, Stefano, and Vaccari 2005]. The feature models are trained with SOMs. An improvement of well-known Scale-Invariant Feature Transform (SIFT) using SOM is proposed in [K. Sharma, Jeong, and S.-G. Kim 2011]. Its objective is to reduce computational time for SIFT. SIFT itself can be applied in object tracking or obstacle avoidance for autonomous vehicles. Also, SOMs can resolve pedestrian tracking collision problem, like in [Humphreys and Hunter 2009] by helping to choose the cost functions for neural networks based on the local-specific tracking context: appearance, change of appearance and trajectory.

3.3 Proposed methods

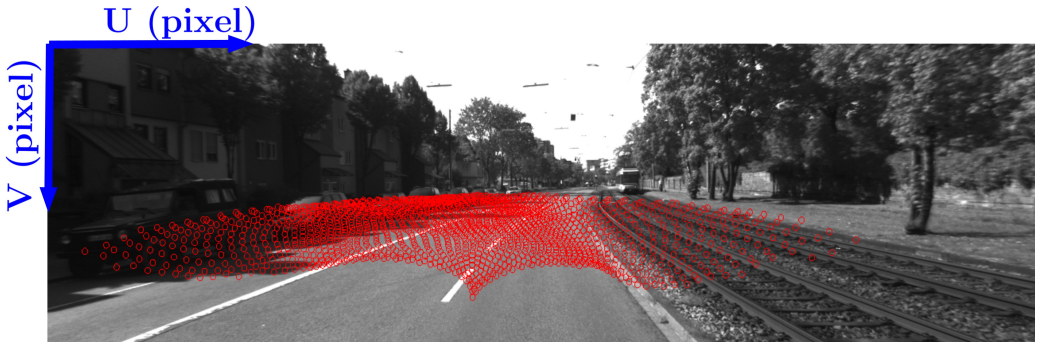
3.3.1 Learning sensor statistics with Self-Organizing Maps

The Self-Organizing Map (SOM) algorithm, originally proposed as a model of cortical information processing, is a generative machine learning algorithm that aims to approximate the distribution of high-dimensional data, and to represent it in a topology-preserving way on a two-dimensional manifold. It is in fact quite related to K-Means [Jain and Dubes 1988] except that the preservation of topology makes it interesting for incremental learning scenarios.

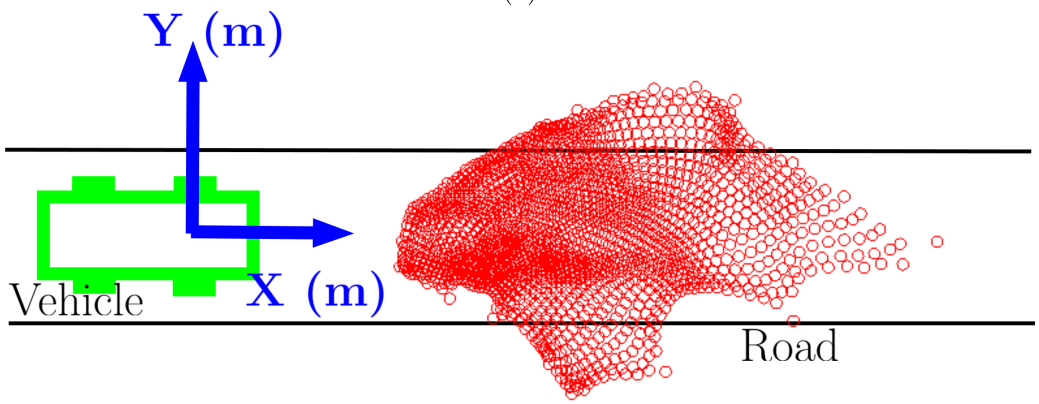
SOM defines a fixed $N \times N$ grid of nodes (neurons) n_i, each of which is associated with a so-called prototype vector $\mathbf{p}_i$, represented an object detection in a detection space. For a given input $\mathbf{z}$, each node gets assigned an activity h_i based on the distance of its prototype to the input:

$$\begin{aligned} h_i &= d(\mathbf{z}, \mathbf{p}_i) \\ d(\mathbf{a}, \mathbf{b}) &= \sqrt{(\mathbf{a} - \mathbf{b})^2} \end{aligned} \tag{3.1}$$

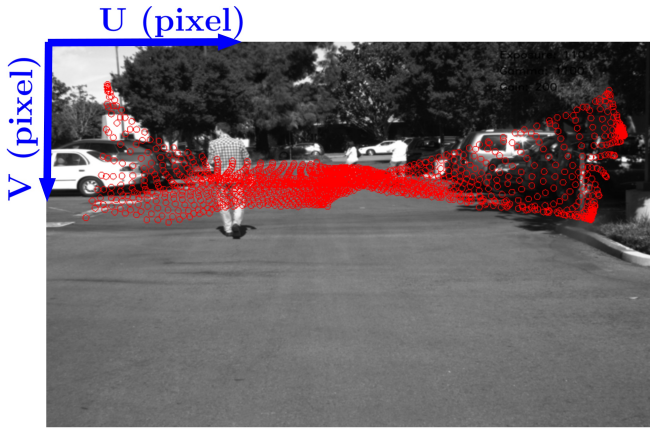
As a distance measure, the euclidean distance is often used, and for our proposal too. In most cases, the calculation of activity is followed by a learning step where the prototypes are adapted



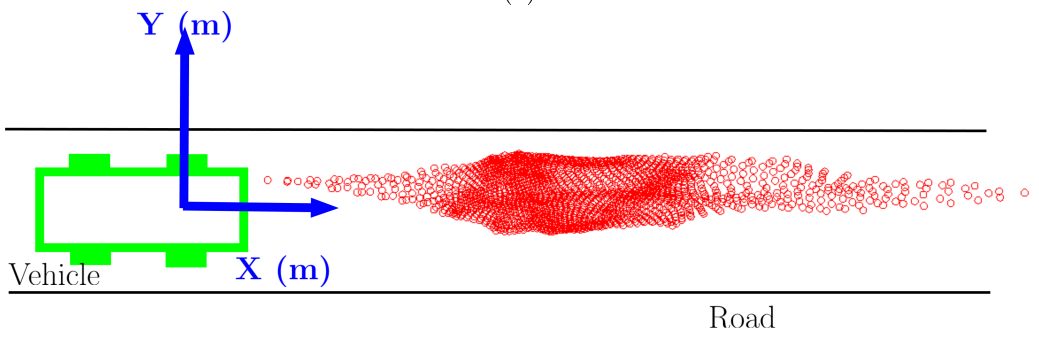
(a)



(b)



(c)



(d)

Figure 3.4 – Statistical models of sensory spaces acquired by self-organizing maps (SOM) for visual (a,c) and LIDAR sensors (b,d). The points represent the position of SOM prototypes in the space of each sensor. The local density of prototypes is guided by average local density of data points. The two datasets-based SOM pairs are presented: (a,b) - SOMs constructed using KITTI dataset, (c,d) - SOMs constructed using datasets from Honda

to better fit the current input:

$$\begin{aligned} i^* &= \underset{i}{\operatorname{argmin}}\, h_i \\ \mathbf{p}_i(t+1) &= \mathbf{p}_i + \epsilon(t) G\left(i^*, i, \sigma(t)\right)\left(\mathbf{z} - \mathbf{p}_i\right) \end{aligned} \tag{3.2}$$

where $G(i, j, \sigma = \exp\left(-\frac{d^2(i,j)}{2\sigma^2}\right))$ is a Gaussian with standard deviation σ which is based on the euclidean distance between node i and node j on the two-dimensional grid of nodes. t is a learning iteration index. For faster convergence, the algorithm demands to gradually lower the learning rate $\epsilon(t)$ and their neighbourhood radius $\sigma(t)$ from initially large values $\epsilon_0,\ \sigma_0$ until the minimal values $\epsilon_\infty,\ \sigma_\infty$ are reached.

3.3.2 Learning of conditional distributions between sensors

Supposing the SOMs are trained using the algorithm described in Sec. 3.3.1, correspondences between visual and LIDAR SOMs are detected using a simple probabilistic counting approach. Assuming that two sets of weights $\omega_{ij}^{\mathcal{L}}$, $\omega_{ij}^{\mathcal{V}}$ exist between nodes i, j in visual $\mathcal{V}$ and LIDAR $\mathcal{L}$ SOMs, both are updated as follows:

for each simultaneously presented pair of visual and LIDAR measurements $\mathbf{z}^{\mathcal{V}},\ \mathbf{z}^{\mathcal{L}}$:

$$\begin{aligned} \tilde{h}_i^{\mathcal{X}} &= \begin{cases} 1 & \text{if } i = \underset{k}{\operatorname{argmin}}\, h_k^{\mathcal{X}} \\ 0 & \text{else} \end{cases} \\ \tilde{h}_i^{\hat{\mathcal{X}}} &= \begin{cases} 1 & \text{if } i = \underset{k}{\operatorname{argmin}}\, h_k^{\hat{\mathcal{X}}} \\ 0.5 & \text{if } i \text{ is neighbour to } \underset{k}{\operatorname{argmin}}\, h_k^{\hat{\mathcal{X}}} \\ 0 & \text{else} \end{cases} \\ \omega_{ij}^{\mathcal{X}} &= \omega_{ij}^{\mathcal{X}} + \tilde{h}_i^{\mathcal{X}} \tilde{h}_j^{\hat{\mathcal{X}}} \end{aligned} \tag{3.3}$$

with a shorthand notation $\mathcal{X} = \mathcal{L}, \mathcal{V}$ ($\hat{\mathcal{X}}$ denoting the other modality is used, i.e., $\mathcal{L}$ if $\mathcal{X} = \mathcal{V}$ and $\mathcal{V}$ otherwise).

After a sufficient amount of samples has been processed, the weight matrices are normalized in order to obtain normalized probabilities:

$$\begin{aligned} \Sigma_i^{\mathcal{X}} &= \sum_j \omega_{ij}^{\mathcal{X}} \\ \omega_{ij}^{\mathcal{X}} &\rightarrow \frac{\omega_{ij}^{\mathcal{X}}}{\Sigma_i^{\mathcal{X}}} \end{aligned} \tag{3.4}$$

It must be noted that the visual and LIDAR measurements do **not** need to come from the same objects. Indeed, if this is the case, it would mean that the desired correspondences to identify are already known. When working on a benchmark dataset like KITTI, this is the case but when training the system on recorded data does not contain any annotations, the correct correspondences are evidently unknown except when there is always just a single object in sight. Therefore, the adopted strategy is: to present all *combinations* of visual and LIDAR

measurements taken at a certain point in time (e.g. a single, synchronized image and LIDAR recording) when learning weights between sensors. This assumes there is a sufficient amount of training data, because the "correct" correspondences will appear together far more often than random incorrect ones. The adopted strategy is applied for two used datasets: KITTI and Honda.

As it was supposed that SOMs have already converged, the SOM learning is disabled during the whole phase of learning conditional distributions by setting $\epsilon(t) \equiv 0$ for both SOMs.

3.3.3 Overall training procedure

The overall training procedure is given in Alg. 1. It consists of a SOM training step and a step that determines weights that have the semantic of the conditional probabilities between the SOM representations of both measurements.

Algorithm 1: Model Training: *Overview over the two-stage model training procedure consisting of learning distributions with SOMs, and learning multi-sensory conditional probabilities.*

for $t : 1 \rightarrow T_{SOM}$ **do**
  Get a random image frame q from D_{train};
  Get random visual measurement l $\mathbf{z}^{\mathcal{V}}_{ql}$ from q;
  Get a random LIDAR frame r from D_{train};
  Get a random LIDAR measurement m $\mathbf{z}^{\mathcal{L}}_{rm}$ from r;
  Update visual SOM with $\mathbf{z}^{\mathcal{V}}_{ql}$ acc. to Sec. 3.3.1;
  Update LIDAR SOM with $\mathbf{z}^{\mathcal{L}}_{rm}$ acc. to Sec. 3.3.1;
Disable learning in SOMs by setting $\epsilon(t) \equiv 0$;
for $t : 1 \rightarrow T_{corr}$ **do**
  Get a random frame q from D_{train};
  for $(l, m) =$ *all permutations of measurements* **do**
    Feed visual SOM with $\mathbf{z}^{\mathcal{V}}_{ql} \rightarrow h^{\mathcal{V}}(t)$;
    Feed LIDAR SOM with $\mathbf{z}^{\mathcal{L}}_{qm} \rightarrow h^{\mathcal{L}}(t)$;
    Update $\omega^{\mathcal{L}}_{ij}$, $\omega^{\mathcal{V}}_{ij}$ acc. to Sec. 3.3.2;
  Normalize $\omega^{\mathcal{L}}_{ij}$, $\omega^{\mathcal{V}}_{ij}$ acc. to Sec. 3.3.2;

3.3.4 Unimodal detection of correspondences

After training is completed, the model is used for detecting whether a given combination of visual and LIDAR measurements is likely caused by the same object. To this end, a criterion is developed. This criterion depends on a single parameter, the probability threshold θ. Assuming that each measurement has generated activities $h^{\mathcal{X}}_i$ in both SOMs, the criterion first computes a single binary measure $c^{\mathcal{X}} = \{0, 1\}$ for each conditional probability matrix $\omega^{\mathcal{X}}_{ij}$, using the

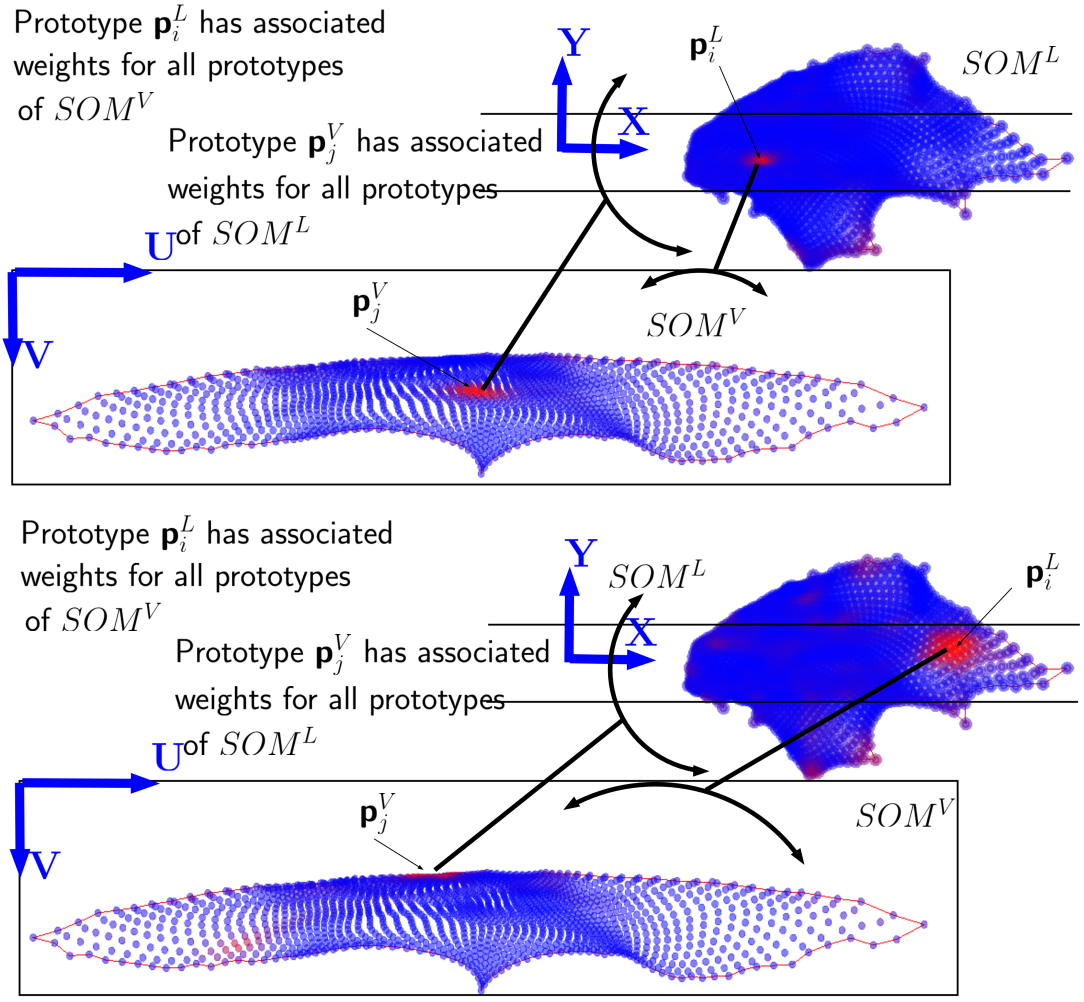


Figure 3.5 – Examples of conditional probability distributions P^{X} for vision (given a LIDAR node) and LIDAR (given a vision node). These distributions are used to detect correspondences. The color of nodes means the value of the association probability: blue tints represent low probability and red tints correspond to high probability.

shorthand notation $\mathcal{X} = \mathcal{L}, \mathcal{V}$ for a certain modality, and $\hat{\mathcal{X}}$ for the other one:

$$\begin{aligned} i^* &= \underset{i}{\operatorname{argmax}}\, h_i^{\mathcal{X}} \\ j^* &= \underset{j}{\operatorname{argmax}}\, h_j^{\hat{\mathcal{X}}} \\ P^{\hat{\mathcal{X}}} &= \left\{ j | \omega_{i^* j}^{\mathcal{X}} > \theta \right\} \\ c^{\mathcal{X}} &= \begin{cases} 1 & \text{if } j^* \in P^{\hat{\mathcal{X}}} \\ 0 & \text{else} \end{cases} \end{aligned} \tag{3.5}$$

The two quantities $c^{\mathcal{X}}$ express whether a best-matching unit (BMU) at position i^* in $\mathcal{X}$ can predict the best-matching unit at index j^* in the *other* modality $\hat{\mathcal{X}}$ based on the learned conditional probabilities. Given a best-matching unit in $\mathcal{X}$, θ is used for selecting a set of nodes $P^{\hat{\mathcal{X}}}$ with conditional probabilities that exceed θ. If the BMU of $\hat{\mathcal{X}}$ is an element of the selected set, one can conclude that there is a match and set $c^{\mathcal{X}} = 1$. Thus, the threshold θ governs the strictness of the matching: if it is high, only a small (or empty) set of nodes $P^{\hat{\mathcal{X}}}$ will be selected and the probability of match decreases. On the other hand, if θ is low, the probability of match

increases, up to the point where there will always be a match at $\theta = 0$. As it is often not necessary to detect all correspondences correctly but rather to exclude unlikely combinations, a more relaxed value of θ helps to avoid missed correspondences while still being able of reducing the combinatorial space of correspondences.

3.3.5 Fusing correspondence detection

Apart from the unidirectional mutual sensor activity predictions, one can also use a cross-verified decision for improving the quality of the correspondence. For that, the criterion of acceptance in Eq. 3.5 changes to:

$$\omega^{\mathcal{X}}_{i^*j^*} \times \omega^{\hat{\mathcal{X}}}_{j^*i^*} > \theta \tag{3.6}$$

$$\omega^{\mathcal{X}}_{i^*j^*} + \omega^{\hat{\mathcal{X}}}_{j^*i^*} > \theta \tag{3.7}$$

$$\sum_k \omega^{\mathcal{X}}_{i^*k} h^{\hat{\mathcal{X}}}_k \times \sum_k \omega^{\hat{\mathcal{X}}}_{j^*k} h^{\mathcal{X}}_k > \theta \tag{3.8}$$

where $h^{\mathcal{X}}_i$ is again the activity at node i in SOM $\mathcal{X}$ (which can be LIDAR or vision, whereas $\bar{\mathcal{X}}$ represents the other modality), and the indices i^*, j^* are the indices of the BMU's in both sensor's SOMs. The last Eq. 3.8 takes into account not only the BMU of each SOM, but also its neighbouring nodes plus their learned conditional probabilities.

3.3.6 Training and evaluation data

Figure 3.6 – The KITTI dataset used for the experiments is recorded from a moving car equipped with several cameras, a GPS device and a Velodyne LIDAR device.

For the training stage and the evaluation of the proposed methods, different datasets are used:

— Dataset A is composed of annotated tracklets from the public KITTI benchmark dataset [Geiger, Lenz, and Urtasun 2012] (see also Fig. 3.6).

— Dataset B is composed of raw detections captured from dash camera and four-layer LIDAR on-boarded on an experimental platform. Visual pedestrian detections are obtained with the 'Daimler' detector provided with the OpenCV vision library, and LIDAR detections are based on the cloud clustering.

— Dataset C is a benchmark created specially for this purpose. The dataset is described in Chap. 5 and its using is described in detail in Chap. 6.

From Dataset A: the center positions of objects in 2D image coordinates are employed, and the corresponding 3D laser coordinates are measured by a Velodyne laser scanner (i.e. tracklets). As the height-over-ground of a tracklet's center is often irrelevant for safety applications, a birds-eye perspective is taken and just considers two of the three 3D coordinates, excluding height-over-ground. Due to the synchronized nature of visual and LIDAR recordings in the Dataset A, each tracklet can be assigned a unique visual image and therefore a corresponding LIDAR sweep. All types of objects provided by the dataset A are used, making the total number of considered tracklets equal to 23497. For training the model, 70% of this dataset is used, performing a random split of available tracklets into train and test datasets.

From Dataset B: the center positions of objects in 2D image coordinates are employed as well as the corresponding 3D laser point cluster center positions. Vision-based and LIDAR-based detection are manually associated to obtain a Ground Truth reference. This sequences is composed of pedestrians filmed in 9 short scenarios of about 2 minutes, making a total number of 8613 visual detections, and 5476 LIDAR detections. Due to the small size of this data base, a cross-validation is performed, that is, for each scenario the SOM are trained with 8 other scenarios and tested with the chosen one.

3.3.7 Evaluation

Algorithm 2: Evaluation: *Overview over the evaluation procedure.*

Disable learning in both SOMs by setting $\epsilon(t) \equiv 0$;
for $i : 1 \rightarrow$ *(images in* D_{test}*)* **do**
  Draw image i from D_{test};
  for $(l, m) =$ *combinations of measurements* **do**
    Feed visual SOM with $\mathbf{z}_{il}^{\mathcal{V}} \rightarrow h^{\mathcal{V}}(t)$ Feed LIDAR SOM with $\mathbf{z}_{im}^{\mathcal{L}} \rightarrow h^{\mathcal{L}}(t)$ Generate bin. measures $c^{\mathcal{V}}$, $c^{\mathcal{L}}$ acc. to. Sec. 3.3.4
Plot precision/recall curves

In order to quantify the capacity of the trained model to identify visual/LIDAR correspondences, the evaluation is conducted on the remaining data. In order to prevent the SOMs from being adapted during the evaluation phase, one set $\epsilon(t) \equiv 0$ for both SOMs.

Assuming a trained model (SOMs plus conditional probabilities), all images in the test dataset are processed in a sequential manner. For each image, all combinations of visual and LIDAR measurements are performed and the scores c_L, c_V for each combination are computed. A binary decision on the presence of a correspondence is taken according to Eq. 3.5. As this decision depends on a single threshold θ the influence is analysed with a Receiver Operating Characteristic (ROC) curve by varying θ in the interval $[0, 1]$ and measuring the precision/recall rates.

A schema of the complete evaluation procedure is given in Alg. 2.

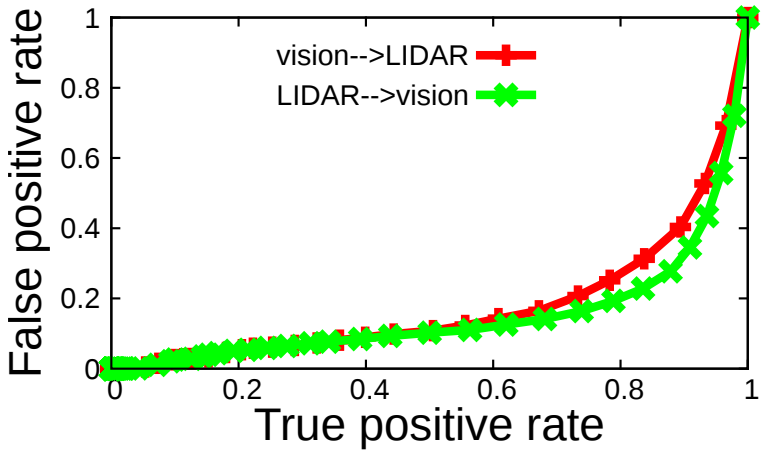


Figure 3.7 – ROCs for vision–>LIDAR (red curve) and LIDAR–>vision (green curve) correspondences. As can be expected, LIDAR–>vision provides slightly better performance as the associated transformation is one-to-one.

3.4 Tests

Model training is performed in two steps.

Step 1: The SOMs are trained independently of one another by drawing random samples from the training dataset, see Sec. 3.3.6, and by adapting each individual SOM according to Sec. 3.3.1, with the input vector provided by the unimodal part of the drawn sample. Training parameters are: $N = 30$, $\epsilon_\infty = 0.01$ $\sigma_\infty = 1$, $\epsilon_0 = 0.6$, $\sigma_0 = \frac{N}{2}$. Neighbourhood radius and learning rate develop according to

$$\sigma(t) = \max\left(\sigma_\infty, \sigma_0 \exp(-\lambda_\sigma t)\right) \tag{3.9}$$

$$\epsilon(t) = \max\left(\epsilon_\infty, \epsilon_0 \exp(-\lambda_\epsilon t)\right), \tag{3.10}$$

with $-\lambda_\epsilon = 0.002$ and $\lambda_\sigma = 0.004$. SOM training duration is limited to $T_{\text{SOM}} = 20000$ iterations.

Step 2: Subsequently, correspondences are trained according to Sec. 3.3.4 for another $T_{\text{corr}} = 20000$ iterations, randomly drawing *images* from the training dataset and feeding all possible combinations of visual/LIDAR measurements to the two SOMs as well as updating the two sets of weights $\omega_{ij}^{\mathcal{V}}$, $\omega_{ij}^{\mathcal{L}}$ based on the resulting SOM activities $h_j^{\mathcal{X}}$, $\mathcal{X} = \mathcal{L}, \mathcal{V}$.

Evaluation is conducted according to Sec. 3.3.7 by iterating over all *images* in the test dataset and measuring precision/recall rates when presenting to the model all possible combinations of visual/LIDAR measurements in each image.

For KITTI base a separate ROC for LIDAR–>vision and vision–>LIDAR correspondence detection is firstly plotted, given in Fig. 3.7. As it can be expected, the LIDAR–>vision correspondence detection gives better results, very likely because multiple LIDAR detections can be associated with a unique vision position. The inverse situation for vision–>LIDAR projection is quite impossible.

It is also noticeable that the algorithm performance is acceptable given that no prior knowledge was used even if it is far from being an ideal ROC.

In a further experiment, it is time to back the claim made in Sec. 3.3.1 that the proposed method was able to handle arbitrary measurements without requiring explicit models of corre-

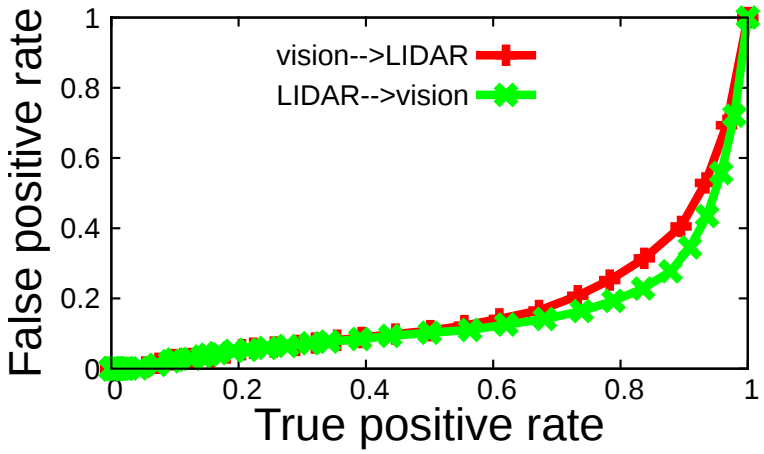


Figure 3.8 – ROCs for vision–>LIDAR (red curve) and LIDAR–>vision (green curve) correspondences, where laser measurements are augmented by object size. By comparison to Fig. 3.7, one may conclude that this irrelevant information is ignored.

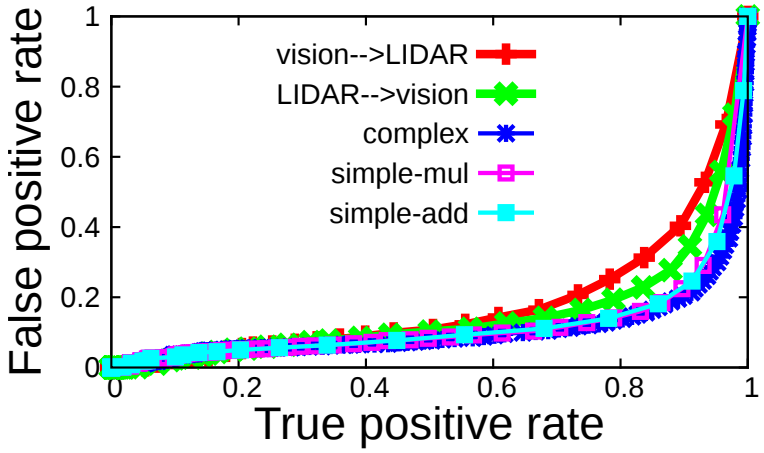


Figure 3.9 – Different ROCs correspondences. The blue "complex" curve represents the cross-verified strategy of Eq. 3.8. The cyan "simple-add" curve corresponds to Eq. 3.7, and the violet "simple-mult" one to Eq. 3.6. It is apparent that all fusion methods outperform the unimodal ones (red and green curves).

spondence. To this end, the previous experiment is realised by taking into account the tracklet width and tracklet height from the laser measurement, bringing up its dimensionality to 4. The ROCs obtained in this way are shown in Fig. 3.8. It is clear that the addition of additional information does not impair the ability of the system to detect correspondences. On the other hand, performance is neither improved, because the added information is irrelevant to the transformation to be computed. This experiment therefore shows that this model, due to the learning approach, is able to process very diverse types of measurements, and automatically extracts the information required for finding correspondences.

Lastly, the three fusion strategies proposed in Sec. 3.3.5 are evaluated, which means that for a pair of visual and LIDAR measurements, there will now be only one decision on correspondence for a given strategy ("complex", "simple-mul","simple-add"), not two as in previous experiments, corresponding to LIDAR–>vision and vision–>LIDAR. The overall performance is shown in Fig. 3.9 and shows that the fused decisions outperform any single unimodal one, boosting the satisfactory performance even further.

For Dataset B (Honda) the ROCs are calculated using only complex activity predictions

from Eq. 3.8 as being the most effective. The results are seen in Fig. 3.10. One can observe a low quality for unidirectional correspondences detections and a very high quality for fused one. It can be explained by the non-symmetrical detections nature and the small number of detected objects per frame. The ROC curves in Fig. 3.10 are placed in order: from left to right, from up to down. In the following it is provided a brief description and analysis of the investigated scenarios:

1. A group of pedestrians (up to 6 people) randomly moves inducing some occlusions. 240 video frames length. In the obtained results, the crossed-validation data association strategy outperforms uni-modal strategies.
2. Two pedestrians move together hand-by-hand. 97 video frames length. The proximity of the walking pedestrians constitutes a non-trivial situation for the association mechanism. As expected, the results evidence a slight decreasing on achieved performance with respect to the other scenarios.
3. Two pedestrians moves describing a crossing trajectory. Their motion is perpendicular to the camera. One occlusion situation is observed. The gap between the pedestrian trajectories is large (depth distance). 447 video frames length. On this occlusion scenario, it was not observed any considerable impact on the global association performance.
4. Two pedestrians move in opposite directions as in the previous scenario. However, the gap between the pedestrian trajectories was small. 183 video frames length. The accuracy observed was considerably decreased with respect to the previous scenario. Since the distance between pedestrian trajectories is small, object positions are harder to be associated.
5. Two pedestrians are crossing the road following the same direction, perpendicularly to the camera. 297 video frames length. Association errors induced by an long-time occlusion were observed in this scenario.
6. Two pedestrian cross the camera's line of sight. The pedestrians trajectories are perpendicular to each other. Only one occlusion is observed. 148 video frame length. No performance changes were observed regarding previous occlusion situations (scenarios 3 and 4).
7. Four pedestrians move randomly with multiple occlusions. This is considered as the more complex scenario. 899 video frames length. The results observed on this test confirm the stability and reliability of this approach. A smooth ROC curve was achieved thanks to this long length sequence.
8. Only one pedestrian is presented. The ROC curve is the best among the reported tests, but not perfect. Errors are frequently caused by the noisy detections employed at the SOMs learning stage. 684 video frames length.

3.5 Discussion and conclusions

A learning approach is presented to solve the problem of finding visual/LIDAR correspondences and to validate its performance on a widely accepted benchmark dataset. In this section,

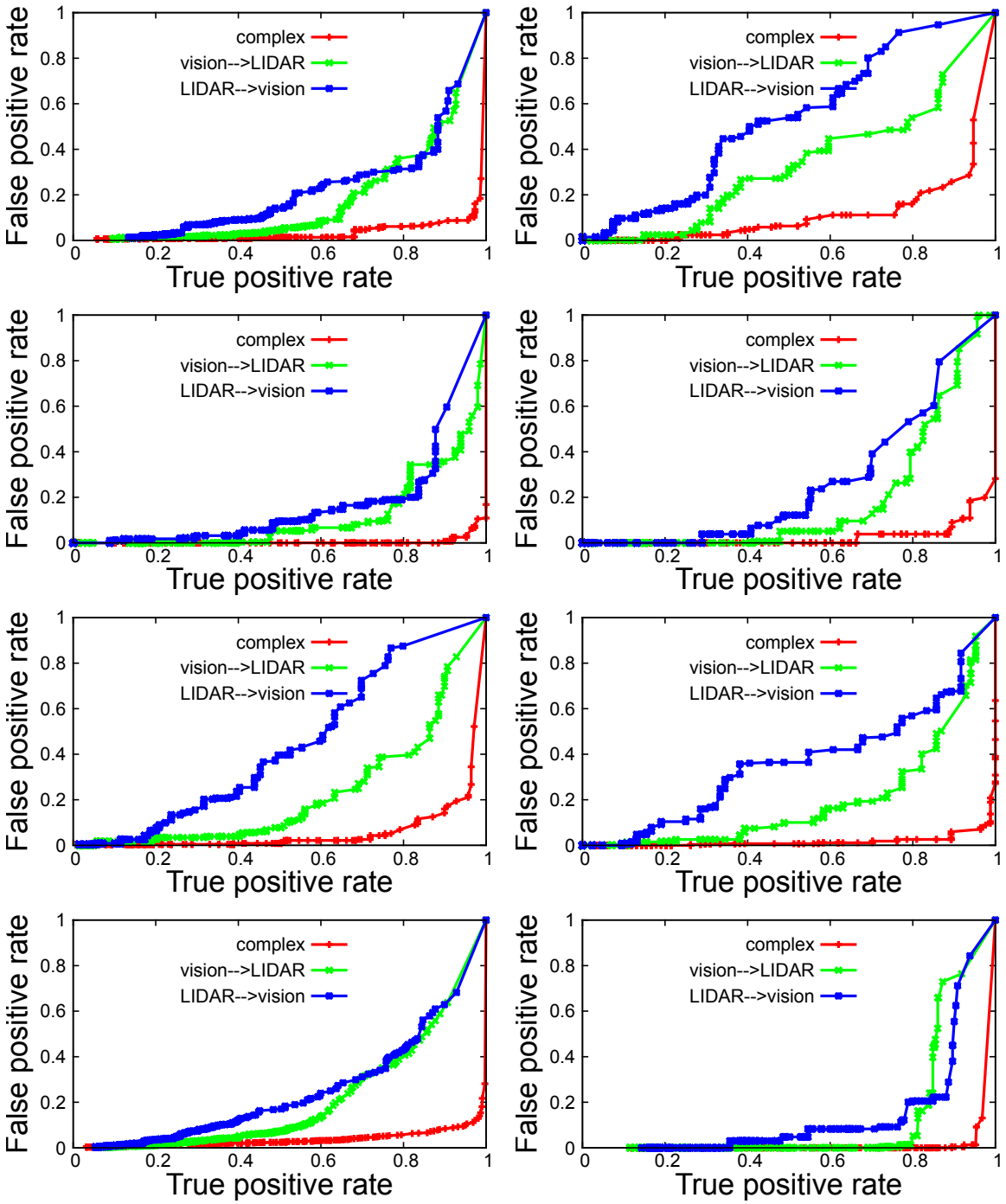


Figure 3.10 – ROCs for fused correspondence detection in the case of cross-validated strategies for 8 scenarios of dataset B (Honda). The red "complex" curve represents the cross-verified strategy of Eq. 3.8. unimodal ones (blue and green curves)

one may review and justify the components of the model and outline the principal conclusions and further research works.

The proposed hybrid SOM-based architecture is based on two necessities: firstly, to have a generic model that will work with any kind of visual/laser measurements. This means that the model must be able to work regardless **what** is actually measured by each sensor. For a camera, this could be, e.g., pixel position of interest points, but also center position, size and identity if an object detection algorithm is used, or center position, size and speed if tracking is added. By using the self-organizing map architecture, every measurement is down-projected to a 2D image-like representation in a way that is statistically optimal and respects a certain topological constraint that allows to easily visualize and interpret a SOM's activity. For ensuring statistical optimality, a variant of the SOM model that has a well-defined energy function [Heskes 1999] is used, which makes very easy to detect measurement outliers that should be ignored.

Secondly, it is desired to have a model that will not fail even when the transformation between modalities is not one-to-one in both directions. To this end, a purely probabilistic approach is adapted on top of the SOM mechanism, that will simply respond by a multi-peaked probability distribution in case where there is inherent ambiguity due to non-unique transformations.

As seen in Sec. 3.3.4, the quality of correspondence finding is very satisfactory given that one did not bring *any* specific expert knowledge. In addition, the threshold θ allows to smoothly change the behavior of the system, from a point where there are few correct correspondences but also few incorrect ones, to a point where there are many correct correspondences but also some incorrect ones. For example, for a multi-modal tracking system a higher false positive rate can be acceptable if no correspondence are incorrectly rejected, since tracking can take into account past information and thus correct the occasional incorrect correspondence. Another very encouraging fact is that the quality of correspondence detection can be significantly improved by considering not only both unidirectional correspondences in isolation, but a fusion of both. As a proper fusion should be, it is indeed better-performing than any single contribution to it.

It was shown that a learning-based approach can successfully solve the problem of multi-modal correspondence detection, in particular between visual and LIDAR sensors. The only prerequisite is a collection of (unlabeled) data which is usually easy to obtain. No expert effort is required at all, and in particular no detailed models of the data acquisition process by the used sensors is needed. The technique is very computationally efficient, and consumes no significant computational load, thus making it suitable for embedded operation. Still it would be better to make this technique even more appealing, for instance by differently attributing the weights to SOM nodes and by better performing fusion strategies.

Chapter 4

Context

Contents

4.1 Introduction

After more than 30 years of contributions on Multiple Target Tracking (MTT), this subject still remains open since, depending on the applications, it addresses complex problems such as management of multiple hypotheses, data association between multiple information sources, and real time constraints. In the context of Intelligent Vehicles (IV), MTT is a key perception process attempting to determine the (e.g. kinematic) state of observed objects. This information is not only important for active safety applications, such as Advanced Driver Assistance Systems (ADAS), but also for scene understanding in autonomous vehicles.

Classic MTT approaches are defined by a recursive framework where a set of detected objects is managed by means of temporal filtering such as Kalman or particle filters. Filtering can usually cope with detection errors and simple missed detections. Multiple Hypothesis Tracking (MHT, [Blackman 2004]), and Joint Probabilistic Data Association Filtering (JPDAF, [Habtemariam et al. 2013; Jaechan 2006]), are part of well-known mechanisms improv-

ing the performance of tracking for complex object-to-track association cases in the presence of missed and false detections.

Temporal filtering contained in MTT frameworks exploits motion models and observed measurements for maximizing the probability of the observed motion. Interactive motion models take advantage of multiple expected high-level motion classes, such as lane changes or turns or stops at crossroads in application to IV contexts [Z. Hu et al. 2012; Cheng and T. Singh 2007]. Those models use online information about recent vehicle motions to predict their future positions. In this study, no motion model classes have been defined, but low-level primitives have been integrated in the form of expected velocity vector fields. Such vector fields are defined by road and lane context, which is taken from maps using ego-localization information [Sattarov, Sergio Alberto Rodríguez Florez, et al. 2014].

Most of the state-of-the-art methods, which exploit context information, are strongly correlated to a particular detection method. For instance, road detection approaches [Strygulec et al. 2013; Chapuis, Aufrere, and Chausse 2002; Ulmke and Koch 2006; X. Hu, Sergio Alberto Rodríguez Florez, and Gepperth 2014] are used to provide key information for driving assistance applications, or to define regions of interest for object tracking [Chapuis, Aufrere, and Chausse 2002; Orguner, Schon, and Gustafsson 2009].

In contrast, this study is inspired by [Maggio and Cavallaro 2009; Kooij, Schneider, and Gavrila 2014; Shibata, Sugiyama, and Wada 2014; Orguner, Schon, and Gustafsson 2009] and aims to use extracted context (road) information to directly improve the quality of multi-object tracking. The contribution in this part is a computational mechanism for integrating *a priori* knowledge derived from contextual road and lane information into a state-of-the-art multi-object tracking system. The benefits of this approach are evaluated in terms of track continuity and track overlapping.

4.2 Proposed methods

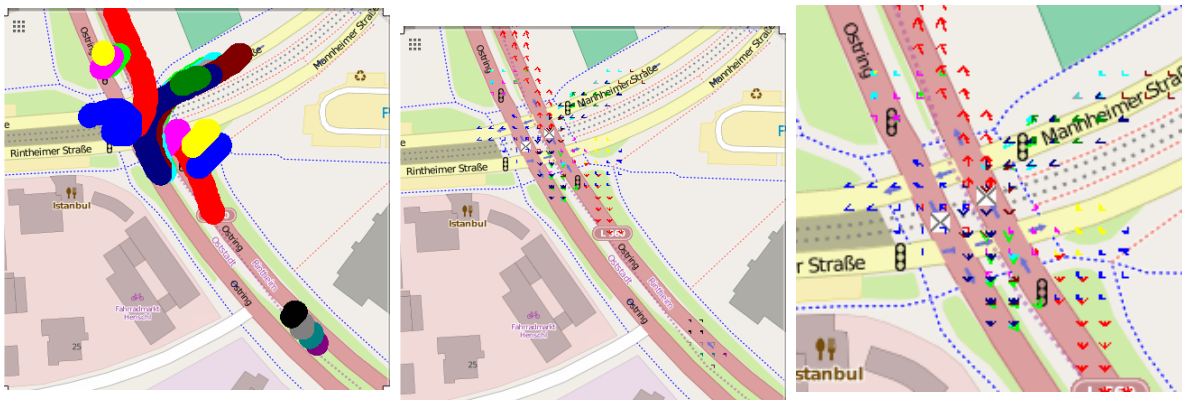
The goal is to track vehicles, pedestrians and other possible objects in two-dimensional space (top-view) while taking scene context into account. Two use cases are considered. **First case:** Simulated scenarios on a featureless 2D map plane with hand-crafted velocity vector field as illustrated in Fig. 2.3

Second case: 2D East-North map space as shown in Fig. 4.1 taken from the KITTI-dataset, from which object information and GPS coordinates serves to match with road and lane context from OpenStreetMap.

4.2.1 Vector field implementation

The context information to implement in the tracking system is represented as a vector field, that is, a field of probable directions for each map location. If $\mathcal{P}$ is the state space of tracking having dimension D, and a subspace $\mathcal{T} \subset \mathcal{P}$ is a space where vector fields are defined, then one point $\tau \in \mathcal{T}$ contains a set of N^τ vectors V in it. One vector $\mathbf{v} \in V$ has components v_i, $i \in 1..D$.

Orientation and norm influence



(a) Map with areas covering by vector fields

(b) Field's vector directions

(c) Field's vector directions, zoom-in

Figure 4.1 – Visual representation of vector fields on a OpenStreetMap (OSM) map

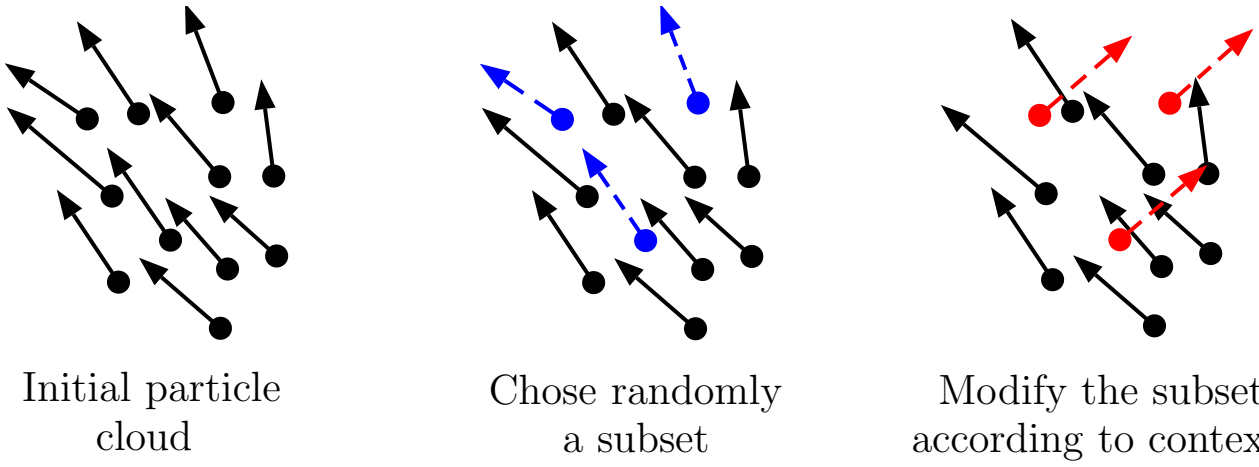


Figure 4.2 – A general idea of particle-level mechanism of external information injection

A track's coordinates c_i, $i \in 1..D$ indicate a point of the tracking space $\pi \in \mathcal{P}$. At the resampling and association stages of PHD filter described in Sec. 2.3.1, when random fluctuations at point $\pi \in \mathcal{T}$ are needed, the vector field is applied.

In a point π it is possible to allow multiple possible context typical directions. Let denote a possible direction π_j, $j \in 1..N^\pi$. The number of particles injected according to the direction π_j is denoted as N^{π_j}.

Let C_{MF} value be the "model force" coefficient. Then the $N \times (1 - C_{MF})$ first particles are resampled as in Eq. 2.27, and other particles are resampled according to vectors, defined in π as follows:

$$\begin{aligned} c_{i,k} &= c_{i,k|k-1} + \zeta_c \\ d_{i,k} &= d_{i,k|k-1} + \zeta_d \\ v_{i,k} &= v_{\pi_j,i} + \zeta_v \end{aligned} \tag{4.1}$$

where $\mathbf{v}_{\pi_j}$ is a vector defined at the point π, with components $v_{\pi_j,i}$, $i \in 1..D$. ζ_c, ζ_d, ζ_v are values defining the evaluation of track's components according to the context. All $N \times C_{MF}$

field-defined particles are divided between vectors $\mathbf{v}_{\pi_j}$, $j \in 1..N^{\pi}$ uniformly. Here N^{π} is a number of vectors defined in π. The main idea is also illustrated at Fig. 4.2.

Direction-only influence

Another potential way to incorporate context consists in letting only the orientation of the vector field influence tracking. In this case, one can calculate new vector components as follows:

$$\hat{v}_{\pi_j,i} = \frac{v_{\pi_j,i} \times \left\|\mathbf{v}_{k|k-1}\right\|}{\left\|\mathbf{v}_{\pi_j}\right\|} \tag{4.2}$$

Here $\left\|\mathbf{v}_{k|k-1}\right\|$ is the norm of the track's speed and $\left\|\mathbf{v}_{\pi_j}\right\|$ is the norm of the vector field's speed at π_j. So, the field's orientation is fused with the norm of the current track's speed. A visual representation of the vector field for the road map is illustrated in Fig. 4.1

Especially the second point is important as it eliminates the need to use vector fields containing all possible speeds (vector lengths).

4.2.2 Vector field compatibility measurement

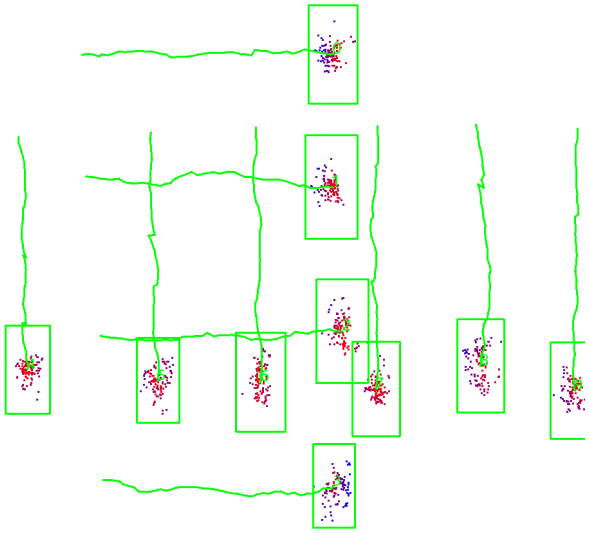


Figure 4.3 – An example of a simulated scenario. The color of particles shows their weight and thus their current impact. The red particles have more weight than blue ones. Green rectangles indicate current tracks.

When the field of possible directions is imposed, it is clear that a moving object may not follow these expected directions. In such a case, it may be assumed that the object has atypical behaviour and is potentially dangerous.

Let use the notation of Chap. 2, where particles weights are denoted $\omega_{x_k,n}$, where x_k is a track k, n is a particle index. We will use the formalism of previous section to identify if a subset of track's particles describes better the track behaviour than the mean of the particle distribution. Therefore we will inject external information in a subset of the particles, denoted ω_{π_j}

The detection of such objects is possible with the proposed framework. If the motion of a tracked object satisfies the following condition:

$$\frac{\sum\limits_{n\in\hat{N}^{\pi_j}} \omega_{\mathbf{x}_k,n}}{\sum\limits_{n=1}^{N^p} \omega_{\mathbf{x}_k,n}} \times \frac{N^p}{N^{\pi_j}} > 1 \tag{4.3}$$

for at least one j, it can be classified as typical. Here, $\hat{N}^{\pi_j}$ is the sets of indexes for particles, resampled according to vector π_j of point π respectively. N^p and N^{π_j} are the number of all object particles and the number of particles resampled according to vector π_j of point π.

The condition considers motion as being "typical" if field-sampled particles are closer to new detections than the mean of all particles. Fig. 4.3 shows a visual distribution of particles' weights. The illustration of the compatibility measurement is provided at Fig. 4.4.

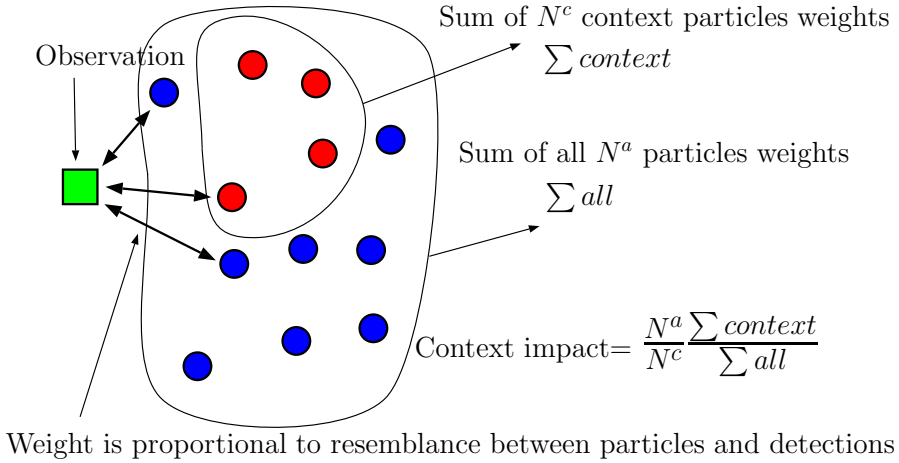
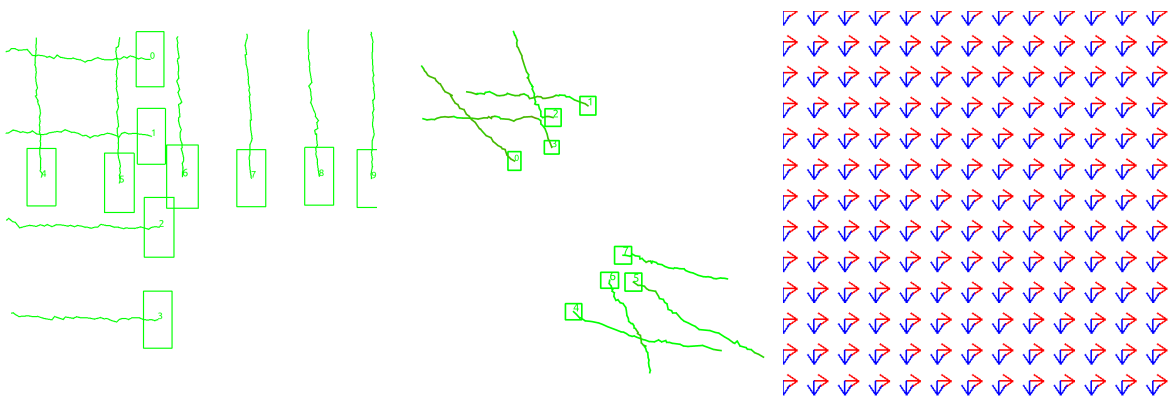


Figure 4.4 – The track's compatibility to the vector field measurements estimation scheme

4.3 Tests

4.3.1 Simulation



(a) (b) (c)

Figure 4.5 – Visual representation of the vector field (c) and simulated scenarios (a,b). Green rectangles and traces are tracks and their previous positions

The first simulation scenario represents a scene of size 1000×1000 pixels and of 110 frames, with 10 objects moving simultaneously: four from left to right, six from up to down as shown in Fig. 4.5a. This scenario is chosen since it contains many pairwise intersections, in order to observe the algorithm's capability to resolve collisions. All of objects have sizes of 30×60 pixels.

Noise parameters were set as follows:

1. $\sigma_d = 10$ - the variance of white noise applied to particle dimensions
2. $\sigma_c = 30$ - the variance of white noise applied to particle centers
3. False negative rate $P_{fn} = 0.1$
4. False positive rate $P_{fp} = 0.2$

PHD imposed parameters are:

1. $R_b = 0.7$
2. $R_d = 0.1$
3. $R_{ret} = 0.1$

The vector field map was created manually and covers all of the scene uniformly with two directions present: "right" and "down" as shown at Fig. 4.5c.

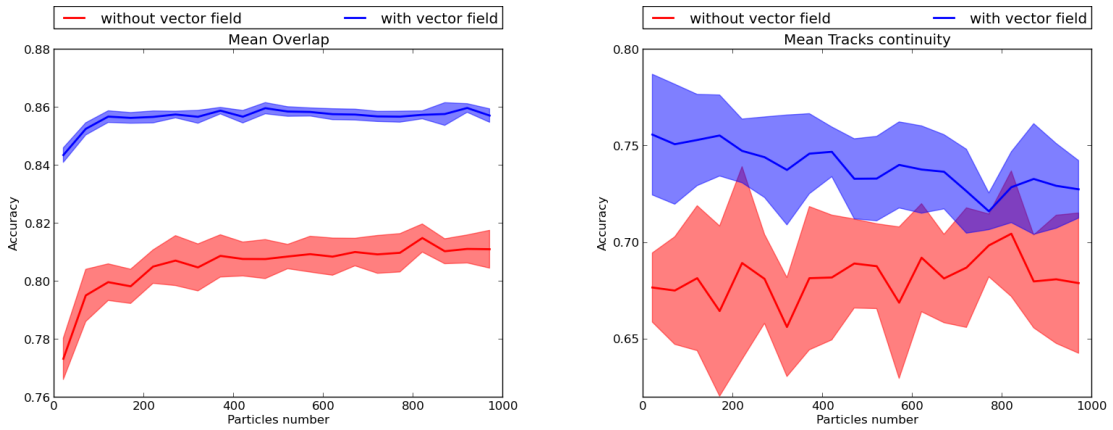
Estimations of overlap and continuity are shown at Fig. 4.6. A larger improvement of the "overlap" is observed while the "continuity" improvement is lower. However both measures are consistently improved by the introduction of the vector field from Fig. 4.5c.

Errors of associations can happen mostly in case of intersections. If two tracks meet at one point, they lose parts of their particle information which can help to resolve the collision because vector fields are identical and come from the same point position. But, on the other hand, if two differently oriented tracks meet in one point, the vector field at this point helps them to go through this point faster. These two reasons balance themselves and so the impact of the vector field is low.

The overlap errors arise from imprecise positions of associations. When the position noise is Gaussian, the trajectories of tracks try to oscillate. When vector fields are applied, the tracks positions become closer to their mean values, and so more stable.

For the same simulated scenario, vector field compatibility measurements were calculated according to Sec. 4.2.2. The mean and standard deviation are measured both for "compatible" and "incompatible" tracks, with the expectation that the compatibility measure allows to distinguish those cases. The compatible tracks were evaluated in the scenario described above, the incompatible ones in a scenario with an inverted vector fields. The results are shown in Fig. 4.7. The difference in mean values is evident, but noise deviations are considerable.

The second simulation scenario represents a scene of size 1000×1000 pixels and of 200 frames, with 8 objects simultaneously: four compatible and four incompatible as it is shown at Fig. 4.5b. All objects have sizes of 15×15 pixels. Noise parameters were set as follows: $\sigma_d = 10$, $\sigma_c = 20$, $P_{fn} = 0.2$, $P_{fp} = 0.125$. PHD imposed parameters are: $R_b = 0.7$, $R_d = 0.1$, $R_{ret} = 0.5$. This scenario will be used further to test the auto-determined model force mechanism.



(a) Overlap (b) Continuity

Figure 4.6 – Accuracy for simulated data when only vector directions are used, plotted as a function of total particle number. Solid lines are mean values, semi-transparent borders represent their variances

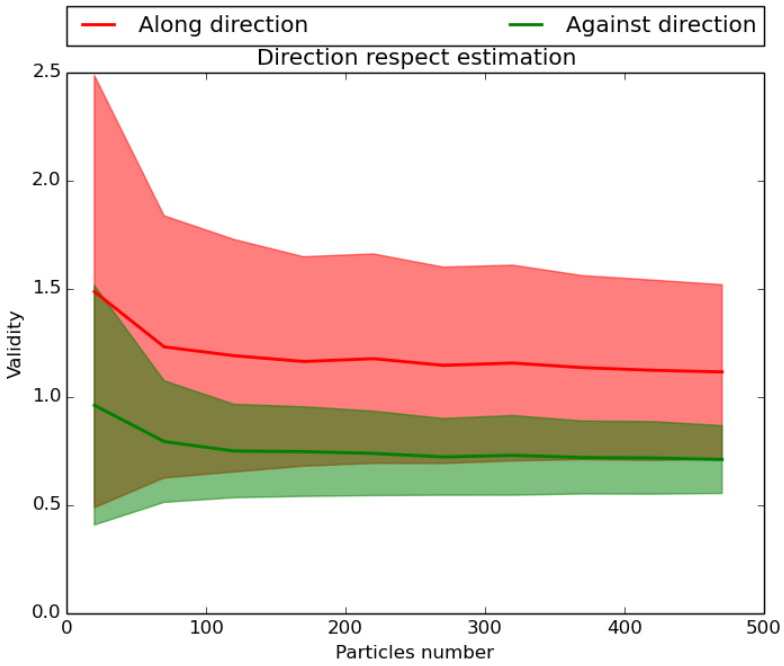


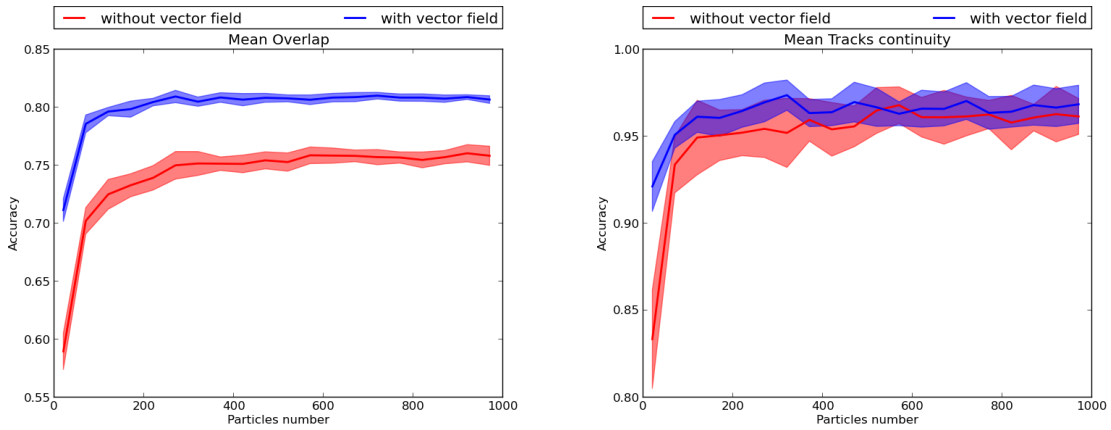
Figure 4.7 – Direction compatibility measurements for simulated data. The values are calculated according to Eq. 4.3. For values bigger than 1.0, the movements is assumed to be along the field, and against the field otherwise.

4.3.2 KITTI/OSM scenarios

The tracking space is a 2D East-North plane limited of size 165 × 167 meters shown in Fig. 4.1. The duration of tracking is 12 seconds with a frequency of 8.9 *fps*. A number of 19 targets takes part in this urban traffic scenario. Since objects like cars, buses, pedestrians and cyclists are present without class distinction, detections of pedestrians can be mixed with detections of cars and other objects.

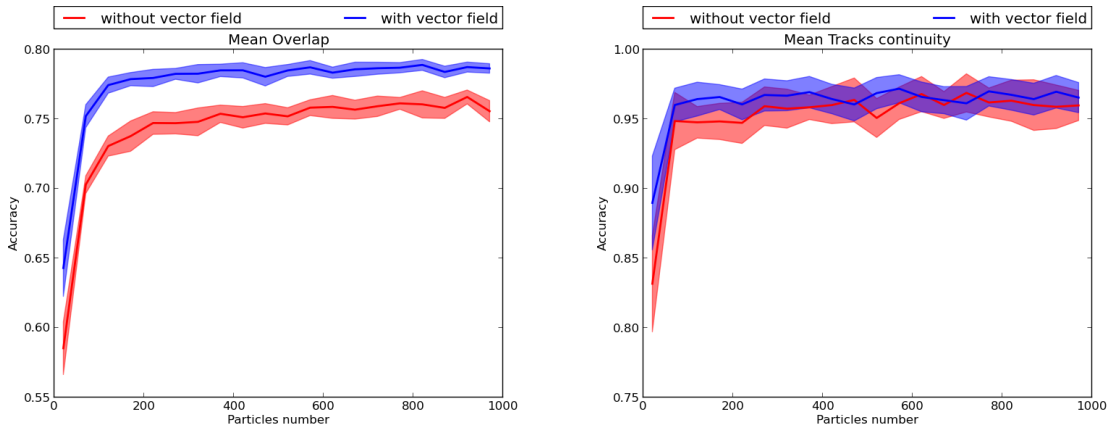
Noise parameters were set to: $\sigma_d = 0$, $\sigma_c = 0.5$ meters, $P_{fn} = 0.1$, $P_{fp} = 0$. PHD imposed parameters are: $R_b = 0.7$, $R_d = 0.1$, $R_{ret} = 0.1$.

The vector field map is created manually with directions collateral to expected target motions in those areas. This field map is based on OpenStreetMap, KITTI Velodyne and GPS-



(a) Overlap (b) Continuity

Figure 4.8 – Accuracy for real data for both vector directions and norms used in dependency of used particles number



(a) Overlap (b) Continuity

Figure 4.9 – Accuracy for real data for only vector directions used in dependency of used particles number

data, and it covers all tracklets' possible occupation areas. The map of directions is displayed in Fig. 4.1b and Fig. 4.1c.

Estimations of overlap and continuity in cases of full information are shown at Fig. 4.84.9. As in case of simulated data, the overlap shows a greater performance difference as a consequence of the vector field. The variance of performance is smaller because of less noise occurring in the real scenario.

We will compare two context introductions: 1) by affecting the track's direction and velocity 2) by affecting the direction only.

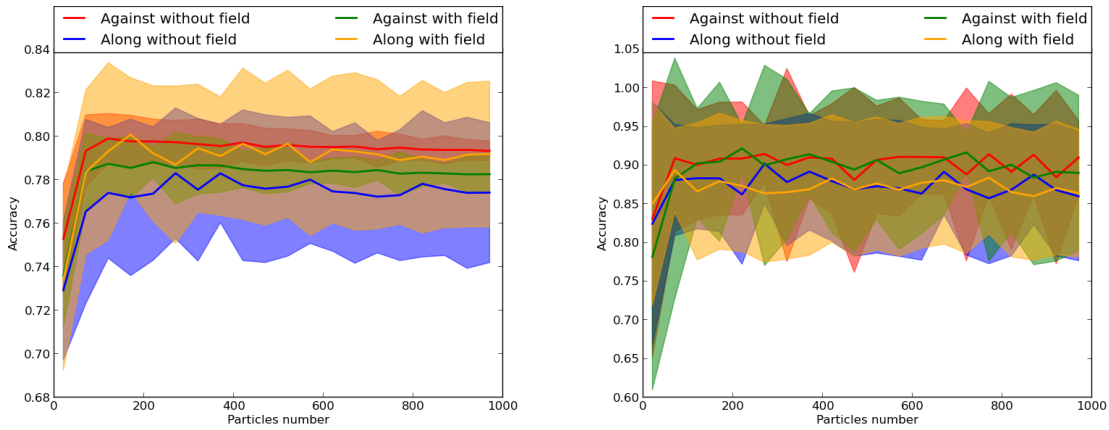
We analyse the performance margins between with- and without vector field. It is possible to draw the conclusion that in this road traffic the complete speed injection is a helpful information, but significant gains in tracking quality can be obtained using only directions.

4.3.3 Auto-determined model force

The second simulated scenario mentioned in Sec. 4.3.1 is created to compare the impact on tracking precision in two cases: 1) movements along vector fields and 2) movements against it.

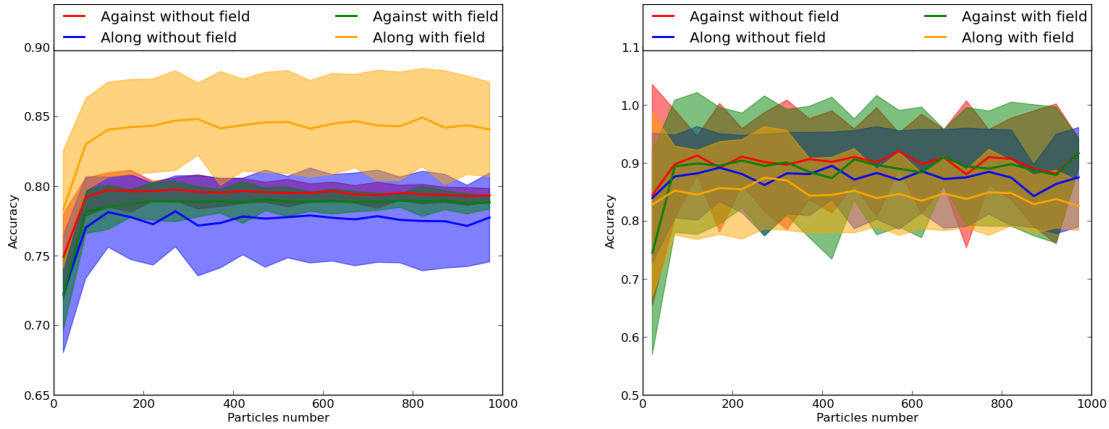
As expected, the results obtained during this experiment show a decreased tracking performance when tracks are incompatible with the context fields. In Fig. 4.10, four lines are shown where blue and red are the respective baselines for compatible and incompatible tracks without the influence of vector fields. Yellow and green curves represent compatible and incompatible tracks assisted by context with a fixed model force coefficient, $C_{MF} = 0.05$. As illustrated, the overlap observed for incompatible tracks is almost the same as the improvement for compatible ones. Track continuity seems not be influenced in both cases.

Hereafter, there is a question of how to keep the advantages of contextual information while reducing the undesirable effects on incompatible tracks. To this end, a dynamic estimation of the model force coefficient, C_{MF}, is proposed. If the track is considered as compatible with respect to the vector field, see Eq. 4.3, C_{MF} is increased by 0.01 or decreased otherwise. For all tracks,C_{MF} varies from 0.01 to 0.5. The results are shown in Fig. 4.11



(a) Overlap (b) Continuity

Figure 4.10 – Comparison of accuracy for tracks moving along and against vector fields without and with them using fixed model force coefficient



(a) Overlap (b) Continuity

Figure 4.11 – Comparison of accuracy for tracks moving along and against vector fields without and with them using a variable model force coefficient

The overlap improvement for compatible tracks clearly outweighs the slight performance decrease observed for incompatible ones. However, track continuity decreases particularly for compatible tracks. This result can be explained in ambiguous tracking situations (e.g. two

tracks intersecting) where contextual information can induce object-to-track association errors. Simulated scenarios contain objects intersecting at the same location with different speed directions. This use case is however not encountered under real conditions.

4.4 Conclusion

In this part, a proof-of-concept of a novel method for multiple target tracking for Intelligent Vehicles is presented. This method uses road information (the context) in order to provide contextual cues which leads to an increased precision in multi-object tracking, using a PHD filter approach in its particle implementation. The public KITTI benchmark dataset was used to verify the impact on tracking precision. The method proves that such kind of *a priori* knowledge is considerably helpful when there is no single *a priori* direction but a distribution over them. The automatic detection of objects that violates the imposed prior is studied with favourable results, promising applicability in safety applications.

Several points are still open, in particular how to correctly encode vector fields (with or without complete speed component).

A subset of Gaussian distribution-initialised particles with modified speed vectors, as used in this implementation, is a possibility, but other distributions, or a more complex particle state including potential high-level behaviors, are conceivable as well.

A principled method to introduce *a priori* knowledge into tracking is presented. In this case information about expected object speeds is obtained from scene context. It is showed, both in a simulated scenario and from the KITTI dataset, that the quality of tracking (measured with overlap and continuity criteria) is significantly improved, leading to a more robust trajectory estimation by a tracking algorithm. Used by different tracking algorithms, the proposed vector field approach can be transferred to all particle-based tracking algorithms and thus has a wide range of applicability.

The subject of object *detection* is not examined: object detections, or Ground Truth, are directly extracted from both the simulated and the KITTI scenario. They are artificially corrupted by noise in order to test the benefits of this approach in noisy environment. Particularly when detections are obtained, as it is envisioned from a object detection processing, our approach is beneficial because the motion *a priori* leads to the best position estimation.

The Gaussian or uniform noise applied to simulate the imprecision of detections are not correct with real databases as most of noise models, and there must be a certain deflection in obtained results regarding to results in case of true noise model, but these noises are simple to implement and frequently used in research applications.

Future work will include a more representative testing using a large set of urban traffic scenarios, to provide more findings regarding the robustness of the proposed methodology.

Chapter 5

Dataset

Contents

5.1 Introduction

Motivation One of the principal problems of multi-modal data fusion research is a lack of reference information, so called Ground Truth, which is needed to quantify the quality of the fusion. Each researcher tries to resolve correctly that problem. For example, one can try to determine with human annotators which results are considered as correctly estimated,

and which one are incorrect [H. Cho et al. 2014]. Such methods cannot be used by another researchers in an objective way, and so methods can not be quantitatively compared. To avoid this problem, reference datasets are often used. Here we enumerate a bibliography of current freely available reference datasets, in order to identify the ones which can be used to evaluate data fusion methods. Particularly, the datasets for multi-modal pedestrian or vehicle tracking in urban scenes were considered in this overview.

Most public datasets in road traffic applications provide labelled Ground Truth for tracking purposes in a single modality. There is a rich class of video tracking datasets, containing for example one of the biggest camera-based datasets for human detection - the Caltech [Dollár et al. 2012; Dollár et al. 2009], or the smaller Multiple Object Tracking (MOT) benchmark [Leal-Taixé et al. 2015], the BoBot tracking dataset [Klein 2010], the Tracking-Learning-Detection dataset [Kalal, Mikolajczyk, and Matas 2012] and many others [Dubuisson and Gonzales 2016].

An interesting, but still multi-camera tracking dataset based on 3D range information is the [Brščić et al. 2013]. A similar multi-camera recording on a large open surface is provided by 3DPeS: 3D people dataset for surveillance and forensics [Baltieri, Vezzani, and Cucchiara 2011] or by S. Sunderrajan's scenarios [Sunderrajan and Manjunath 2013; Sunderrajan and Manjunath 2016].

There are RGB-D datasets, where multi-modality comes from RGB and ToF cameras used together. One of the most commercial known device of such type is the Kinect sensor [Cai et al. 2016]. The used scenarios in these datasets are filmed in closed rooms, but the integration in intelligent vehicles is a currently ongoing activity. RGB-D solutions have their principal application as autonomous robots perception system [Sturm et al. 2012]. As another reason not to use ToF-based datasets is the pre-combined data fusion with joint RGB-D output.

Two datasets published on the International Workshop on Performance Evaluation of Tracking and Surveillance (PETS) are multi-modal in nature:

— The ARENA dataset [Patino and Ferryman 2014] contains multi-camera recordings for human tracking and complex interactions around a parked vehicle. The cameras are non-overlapping and the vehicle on which the cameras are mounted is static, so it can not serve for the purpose of the algorithms of Chap. 3

— The IPATCH dataset containing 14 overlapping visible and thermal recordings [L. Li, Nawaz, and Ferryman 2015]. The dataset challenges are detection, tracking and scene understanding in the maritime domain.

There is a class of multi-camera based datasets where fixed cameras observe a small limited area full of moving pedestrians. Examples are the "EPFL" [Berclaz et al. 2011; Fleuret et al. 2008], SALSA [Alameda-Pineda et al. 2015], HALLWAY and LAB datasets [T. Hu, Messelodi, and Lanz 2015], VIPT [Mutlu, T. Hu, and Lanz 2013], MVPDT [T. Hu, Mutlu, and Lanz 2013].

The Toyota Motor Europe (TME) Motorway Dataset provide 28 movie clips of highway scenes with vehicle annotations [Caraffi et al. 2012]. The perception data is composed from image stereo acquisition, ego-motion estimate and laser-scanner generated vehicle annotations. The GT annotations are limited to angular resolution of the laser scanner. Furthermore, due the discontinuity of automatically generated GT object trajectories, some tracks are actually processed as multiple tracks from different objects.

A true multi-modal dataset for human motion is proposed by Berkeley Multi-modal Human Action Dataset (MHAD) [Ofli et al. 2013]. It has optical motion capture system, four multi-view stereo vision camera arrays, two Kinect cameras, six wireless accelerometers and four microphones. Unfortunately, the filmed scenarios are not vehicle-perception-based in urban environment.

The Daimler Urban Segmentation Dataset consists of video sequences recorded in urban traffic with stereo image pairs [Scharwächter et al. 2013; Scharwächter et al. 2014]. Dense disparity maps are provided as a reference and annotations are computed using semi-global matching.

The CLEAR dataset contain audiovisual information from multiple acoustic and video sensors [Katsarakis et al. 2008]. It is designed for multi-modal speaker tracking [Taj, Maggio, and Cavallaro 2008]. As most multi-camera sets, the recording is performed in a closed room with fixed cameras and microphones.

The BU-TIV (Thermal Infrared Video) Benchmark provides thermal, multi-view thermal and multi-view color-thermal image recordings for dynamic object tracking [Z. Wu et al. 2014]. The cameras are fixed, and recorded objects are pedestrians inside and outside buildings, as well as vehicles on a road.

The Object Scene Flow dataset is based on Stereo perception data and optical flow [Menze and Geiger 2015; Keller, Enzweiler, and Gavrila 2011]. This is an example of dataset that is derived from KITTI.

The EuRoC MAV is a visual-inertial dataset collected on-board a Micro Aerial Vehicle (MAV) [Burri et al. 2016]. The dataset contains stereo images, synchronized IMU measurements and motion and structure Ground Truth. The dataset is an example of multi-modal perception, but its goal is self-localization and environmental structure modelling.

A set of recordings united under the project name RAWSEEDS represents a mobile robot equipped with GPS, cameras, LIDAR, IMU, odometry sensors etc. indoor and outdoor of a campus [Bonarini et al. 2006; Ceriani et al. 2009]. The aim of the benchmark is to provide data fusion for self-positioning and environment reconstruction purposes. The Ground Truth consists in the robot's actual trajectories.

One of the most well-known urban traffic tracking datasets is KITTI (from Karlsruhe Institute of Technology) [Geiger, Lenz, and Urtasun 2012], [Geiger, Lenz, Stiller, et al. 2013]. It provides high-precision LIDAR-based Ground Truth, convertible to image projections via extrinsic calibration, that means the only one GT modality exists which makes data fusion tasks too artificial. At the same time, the problem of multi-modal perception is strongly coupled to the sensors' different characteristics, mainly expressed by variations in the field of view, which means that sometimes an object which is observed with one sensor is occluded in another's zone of visibility.

The LIPD dataset proposed by Cristiano Premebida [Premebida and U. J. C. Nunes 2009] is close to the task of multi-modal data fusion: it contains synchronous LIDAR and raw camera measurements and their Ground Truth (positive and negative samples for a pedestrian detection and classification task). That dataset also provides encoder and DGPS information for rover position estimation. But still, that dataset is not appropriate for the task of real-time multi-

modal tracking: the Ground Truth is there focused on the multi-modal classification task. In contrast, real sensors have different frame rates, they are not synchronized, and they can detect something not detected by others (in the dataset the area of visibility is the same, and image ROIs are generated with the help of range detections). Finally, the Ground Truth in LIDAR and camera spaces does not provide a Ground Truth in world coordinate systems, when the latter is useful to better understand urban scenes. Finally, LIPD does not describe the temporal continuity of detected pedestrians, which is not needed for the detection but crucial for tracking.

The dataset "A multi-sensor traffic scene dataset with omnidirectional video" provides a benchmark with 360°field of camera view and multiple self-positioning estimation sensors, as GPS, IMU, lateral and longitudinal velocity sensor and three non contact laser ride height sensors [Koschorrek et al. 2013]. Unfortunately, the Ground Truth is not provided and multi-modality here concerns only the position of the rover the data are recorded from.

Commercial "Multisensor Datasets" from VisLab s.r.l. provide also multi-modal vehicle perception, including cameras, stereo pairs, tetravision systems, LIDAR, GPS, IMU and chassis data, but also without GT annotations [Vislab 2011].

The authors of "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments" have recorded multi-modal datasets (based on RGB-D and 2D LIDAR) with separate annotations [Linder et al. 2016]. They also propose their own framework for multi-sensor annotations. The recorded scenarios contain pedestrian crowds in a laboratory and an airport from the point of view of a mobile human-size robot. This benchmark can be considered as one of the nearest to the searching criteria, except that it was not recorded in an urban traffic environment.

In the Tab. 5.1 a short classification of the above-mentioned datasets for multi-object tracking is provided. As can be seen from Tab. 5.1, no publicly available dataset precisely fits our intentions, which is why it was decided to create an own dataset for this purpose.

Our dataset provides four sources of information, three of which have separate and independent annotations. These sources are: LIDAR point clouds (GT from segmentation), RGB cameras (GT from visual pedestrian detection methods), signals from portable GPS-devices and wheel-based vehicle odometry. A presentation of different coordinate system is done in Sec. 5.5.3, followed by the dataset file structure in Sec. 5.5.3. The recorded scenarios are presented in Sec. 5.5.3.

5.2 Scene sensors

5.2.1 Vehicle-embedded sensors

The sensors used to register raw data about the vehicle state and the observed environment are divided on two groups: 1) Vehicle-embedded sensors are devices that are rigidly attached to the vehicle, and 2) track-embedded sensors are devices that are attached to mobile tracks (i.e., carried by pedestrians).

Further, one can use the terms "rover"and "vehicle" as synonyms, defining a vehicle embedding a set of sensors. As we are dealing with pedestrian tracking here, the terms "tracks" and "pedestrians" will often be used in a synonymous fashion as well.

Dataset group	Examples	Multi-sensor	Urban road scenario	Object tracking
Mono-modal tracking	Caltech, MOT, BoBot, TLD etc.	no	yes	yes
Fixed multi-camera with highly intersected view	EPFL, SALSA, HALLWAY, LAB, VIPT, MVPDT etc.	partly	no	yes
Fixed multi-camera (or range) with weakly intersected view	ATC SC, 3DPeS, Sunderrajan's, ARENA etc.	partly	no	yes
Audio-visual	CLEAR	yes	no	yes
Fixed thermal-color view	BU-TIV	partly	yes	partly
Stereo or optical flow	KITTI, Menze's Daimler etc.	partly	yes	yes
RGB-D ToF-based	Cai's,Sturm's etc.	partly	partly	yes
LIDAR-based annotations	KITTI, LIPD, TME etc.	partly	yes	yes
Multi-sensory robots	EuRoC MAV, Rawseeds etc.	yes	partly	no
Without Ground Truth	Koschorrek's VisLab's etc.	yes	yes	partly
No urban scenes	Linder's	yes	no	yes

Table 5.1 – Short dataset taxonomy for multi-sensor object tracking in urban road scenarios

Vehicle-embedded sensors have two objectives: to estimate the rover's world position, such as odometry-served wheel encoders and GPS devices, as well as to observe tracks, such as monocular vision cameras, or LIDAR sensors.

All embedded sensors are connected to a vehicle-mounted PC.

To better evaluate the performance of multi-object tracking, it is useful to use at least two (overlapping) sensors that observe the main part of conductor's view zone. In the standard approach, the front camera as the cheapest solution for object tracking in front of the vehicle is used. In addition to that, it is useful to add a sensor with higher object resolution quality. As a Ground Truth source, one can take an external data source with high precision and the field of view larger than any of rover's embedded sensors.

The finally selected sensors are:

1. A front camera with a large-angle objective. The choice is made since the most important possible actions of tracked objects are expected in front of the vehicle.
2. A LIDAR sensor with four layers oriented in front of the vehicle. Its visibility zone is tighter that of the camera in terms of horizontal angular range. The chosen type of LIDAR is cheaper than a Velodyne sensor and thus potentially usable in consumer vehicles. At the same time, its lateral resolution is more precise than that of the camera.
3. GPS antennas on tracked objects are chosen to provide Ground Truth information.
4. In order to add some additional information that might prove useful to the data fusion task, odometry information is also provided.

A schematic Fig. 5.1 shows the main elements for dataset recording: a rover with installed sensors, a GPS base for RTK-precision processing and a reference pedestrian equipped with Raspberry-Pi [Molloy 2016; Halfacree and Upton 2012] device with a GPS receiver.

The camera installation on the bar of rover's top is shown in Fig. 5.3b.

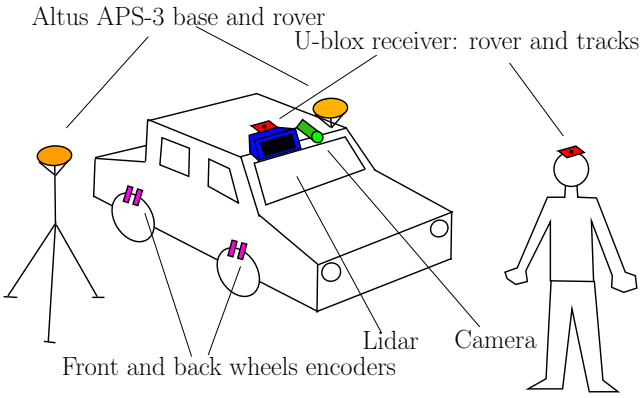


Figure 5.1 – Dataset recording elements schema

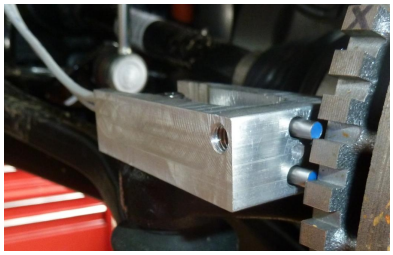
Both for track control and for configuring the GPS station, as well as data collection from rover sensors, a PC with a dual-core Intel Core processor and hard disk storage capacity of 250 Gigabytes was used. The computer runs Ubuntu Linux (64 bit).

Odometry

Odometry is a measurement of travelled distance as recorded by the rover itself, providing a continuous, but potentially drifting motion estimate relative to some arbitrarily fixed initial location.

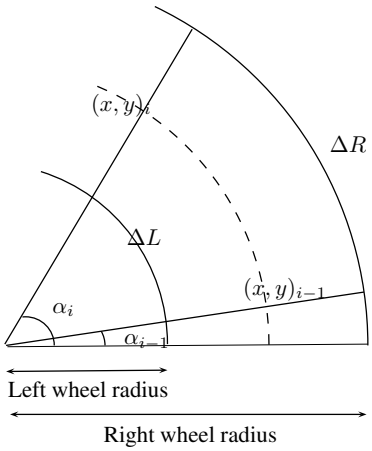
A so-called Wheel Speed Sensor (WSS) is a type of speedometer, i.e., a sender device used for reading the speed of a wheel rotation.

Inductive sensors used in the rover are Contrinex DW Ax-62x-04 devices [*Inductive sensors* 2013] of 4mm diameter. Their maximal frequency is 3000 Hz, so 230 km/h (computed using 90 "teeth").



(a) Encoder with double sensors installation

(b) Detection track on the brake drum



(c) Odometry calculation schema

Figure 5.2 – Odometry sensors. Images (a) and (b) are taken from [Bouaziz 2013]

Four Rotary encoders are placed in the rover's wheels. Two front wheels have quadrature encoders, which are able to determine rotation direction with an angular resolution of 1° (90 teeth for 4 combinations of two sensors). The two rear wheels have a single sensor each, which are not able to determine rotation direction (only rotation speed), working at an angular resolution of 4°. A circle with a detection track installed on a wheel is shown in Fig. 5.2b. An encoder with two sensors (with quadrature signal) installed on the rover is shown in Fig. 5.2a.

The motion model of the rover is composed of a pair of wheels with given radius R (m) and distance between wheels D (m). The rover's state is parametrized by its position as the center point between the wheels, and its yaw angle between a current motion direction and an initial one.

At each new observation, the new rover's coordinates x_{i+1},y_{i+1} in meters and rotation angle α_{i+1} in radians are updated from old values x_i,y_i,α_i according to Eq. 5.1

$$\begin{aligned} \alpha_{i+1} &= \alpha_i + \frac{\Delta L - \Delta R}{D} \\ x_{i+1} &= x_i + \frac{\Delta L + \Delta R}{2} \times \cos \alpha_{i+1} \\ y_{i+1} &= y_i + \frac{\Delta L + \Delta R}{2} \times \sin \alpha_{i+1} \end{aligned} \tag{5.1}$$

where distance increments in meters for left and right wheels respectively ΔL (m) and ΔR (m) are calculated from impulse increments δL and δR with impulses per turn n of rotary encoders and wheel radius R:

$$\begin{aligned} \Delta L &= \delta L \times \frac{2\Pi R}{n} \\ \Delta R &= \delta R \times \frac{2\Pi R}{n} \end{aligned} \tag{5.2}$$

In Eq. 5.1, the angular increment is supposed to be small and so it is approximated to $\sin \Delta\alpha \approx \Delta\alpha$ [Olson 2004]. The schema with odometry parameters is illustrated on Fig. 5.2c.

Of course the CAN messages from encoders arrive at different times, but when they are processed in the odometry equations above, four consecutive messages from all four encoders are considered to be simultaneous.

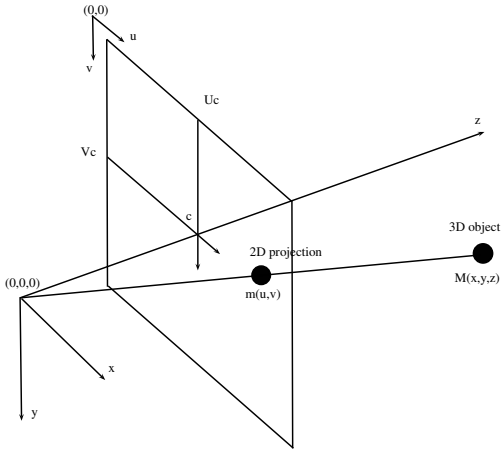
Monocular vision

The camera is meant to provide a faithful and full representation of the scene in front of the vehicle. The field of view must not be too narrow, because in that case close, but laterally distant objects will be ignored. Too wide field of view is to be avoided too, because it leads to image distortion and makes tracking problematic. In addition, an adequate resolution in order to detect pedestrians at a maximal distance of 25 meters is needed.

The chosen PointGray Flea2 color camera (model number FL2-08S2C) is fixed on the rover's top. The camera has a bus connection interface IEEE-1394b which allows the 800Mb/s interface speed for full color RGB output. Sensor model is ICX204 1/3" with a resolution of 0.8 Megapixels(1032x776), with sensor's pixel size 4.65x4.65μm, imaging sensor type CCD [*Flea 2 Technical Reference Manual* 2011].

The camera is equipped with lens Theia MY125M with focal length 1.3, aperture 1.8, horizontal angle of view 135°, vertical angle of view 119°, diagonal angle of view 141°, manual locked focus and a resolution of 5 Megapixels.

That camera has a large enough field of view, it can provide color images of schema RGB8 with sufficient frequency of 15 fps and a resolution of 1024x768 pixel. These parameters are set up for all experimental recordings.



(a) Schema of projection from 3D camera space to 2D image space

(b) Camera installation on the rover's top

Figure 5.3 – Camera as visual sensor

The pinhole camera model has an intrinsic parameter matrix **A** which is needed to transform 3D camera space to a 2D plane projection:

$$\mathbf{A} = \begin{pmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.3}$$

where $f_x = f \times m_x$ and $f_y = f \times m_y$ represent focal length in pixels, m_x and m_y the scale factors of pixels to distance, and f the focal length in terms of distance. γ is a skew coefficient between the axes x and y. u_0 and v_0 represent the principal (center) point in the projected image.

A schema of camera 3D space projection into 2D space is represented on Fig. 5.3a.

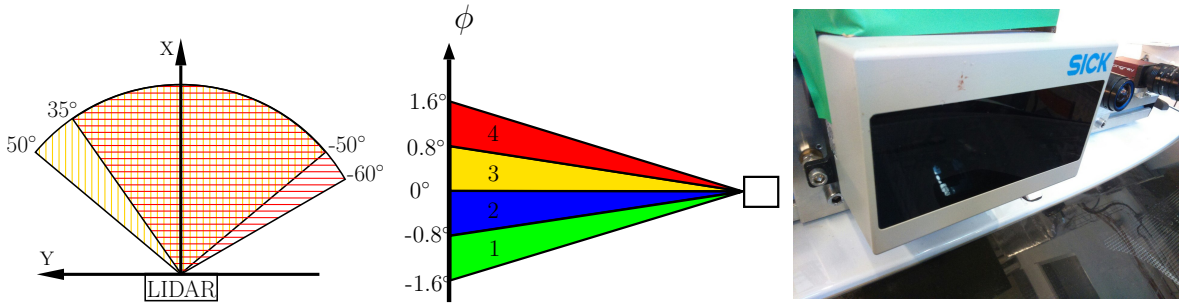
The chosen camera has a large angular field of view, and thus, the provided images have deviations from rectilinear projection on their sides, i.e., distortions that must be taken into account in the camera model. For this case, it includes 4 additional non-zero coefficients: two radial distortion coefficients k_1, k_2, and two tangential distortion coefficients p_1, p_2. After their determination, the camera images can be undistorted as described in Sec. 5.3.2.

Range scanner

The principles of LIDAR sensors are detailed in Sec. 1.2.3.

A 3D Laser Scanner (LIDAR, model number LD-MRS400001) with 4 measuring planes is placed on the rover's top. This LIDAR sensor has 85°of central scanning range and 110°of total scanning range. The illustration of its angular resolution is given in Fig. 5.4a. The

frequency range varies between 12.5 Hz and 50 Hz with different possible horizontal angular resolutions, which might typically take values of 0.125°, 0.25°, 0.5°. The four layers are scan planes, one under another, and have 0.8°of vertical angular distance. The Fig. 5.4b illustrates those values. The LIDAR sensor has an Ethernet data interface [*Laser Measurement Sensor LD-MRS Operating instructions* 2014].



(a) Horizontal angular range (b) Vertical plane angles separation schema (c) View of LD-MRS SICK 3D Laser Scanner

Figure 5.4 – LIDAR as range sensor

The frequency is set to 25 Hz with 0.25°of horizontal angular resolution. One LIDAR scan contains all four layers in the LIDAR coordinate system.

The chosen model LD-MRS SICK is shown on Fig. 5.4c.

Scanning micro-pulse LIDAR sensors get reflections of objects from consecutive times, but one scanning sweep is supposed to be simultaneous because of negligible turning and reflecting time. The LIDAR sensor itself is at the center of its own Cartesian 3D space, in which reflections are modelled as 3D points.

A LIDAR scan is also a subject to deformations induced by the rover motion: while initial ranges in a scan are measured in a previous vehicle's position, the last ranges are from more recent one. Also, the range computation for a full scan takes some time, while rover's position changes. With a scan frequency 25 Hz and for a rover speed limited to 30 km/h, the maximal deformation estimation is about 0.33 meters. It is a full scan linear deformation. The deformations inside one scan are smaller.

5.2.2 GPS localisation

The definition of GPS and its enhancements are described in Sec. 1.2.5.

In this section, the applications of GPS technology for the precise positioning of the rover and tracked objects are studied.

As it is shown in Fig. 5.1, there are two types of GPS devices playing a role in the dataset creation as well as other parts:

1. Altus APS3 is a centimeter RTK accurate smart antenna of Serptentrio manufacturer [*APS-3 User manual* 2011]. Its accuracy is described in Tab. 5.2. The APS 3 is equipped with SIM and SD cards. For control and data transmission, a cable with 5-pin LEMO port of APS 3 and an RS232 port at the other end is present. The Altus APS 3 can store raw satellite data on its SD card at a maximal frequency of 25 Hz. With the same frequency, it can output NMEA string messages via the cable.

Navigation performance	Horizontal (m)	Vertical (m)
Standalone	1.3	1.9
SBAS	0.6	0.8
DGPS	0.5	0.9
RTK	0.01 +1 ppm	0.02 +1 ppm

Table 5.2 – Navigation accuracy for Altus APS 3

2. The U-blox M8 module is a passive GNSS receiver from the u-blox manufacturer equipped with UART and USB output ports [*u-blox 8 / u-blox M8 Receiver Description* 2016]. It is able to ouput, at a frequency of 5 Hz, the NMEA string messages and messages in UBX format, including RXM_RAWX (providing raw measurements from satellites, i.e. pseudoranges and carrier phase measurements) and RXM_SFRBX (providing raw navigation data, i.e. ephemeris, satellite clock and ionosphere parameters) messages. Although u-blox has only one L1 antenna, this is sufficient to use the RTK mode with floating-point precision. The antenna has a 35mm square ceramic patch [*CGGBP.35.6.A.02 Specification datasheet* 2015].
3. The Raspberry-Pi is a credit card-sized single-board computer manufactured by the Raspberry Pi Foundation. The Raspberry Pi 1 Model B+ is used. It is equipped with 4 USB 2.0 ports and a micro-SD memory card. The installed operation system is Ubuntu Linux 64 bit.
4. The Edimax WiFi antenna (EW-7811Un) with 150 Mbps data transfer speed is connected to the Raspberry-Pi device to control it by remote commands.
5. The RS 5200 mAh lithium Power Bank battery pack 775-7508 is used to power the Raspberry-Pi.

(a) The Altus APS 3 smart antenna

(b) The Raspberry-Pi device with U-blox U8 receiver, battery and WiFi antenna

Figure 5.5 – Various GPS receivers

To obtain the most precise GPS data possible, the RTK mode was used. One Altus APS 3 antenna was placed statically on a tripod as an RTK base. The second APS 3 was fixed on the top of the experimental vehicle as a reference RTK rover, as it is illustrated in Fig. 5.1. Both are configured to log raw measurement data internally on SD cards at a frequency of 25 Hz. The rover's APS 3 is also configured to output NMEA strings via its RS232 communication

interface at 25 Hz. This is necessary to associate GPS and POSIX times in order to synchronize POSIX-timestamped camera, LIDAR and encoders with GPS-timestamped receivers as it will be shown in Sec. 5.4.2. All ports are configured to have a baud rate of 115200.

Along with the reference APS 3 rover antenna, a less precise U-blox antenna is fixed on the top of the vehicle to provide perception measurements. U-blox is configured to output via the USB port the UBX messages containing raw data, and to output via the UART port the NMEA string messages to make the association between GPS and POSIX times.

Tracked pedestrians hold Raspberry-Pi blocks connected to U-blox antennas, as well as WiFi antennas powered by lithium batteries. The U-blox receiver is placed on the hat of each pedestrian and is configured to output via USB port the UBX messages with raw GPS observation data. The recording itself is effectuated under Linux.

The RxTools software is a suite of GUI tools for monitoring and configuring receiver operations as well as logging and downloading SBF data files. There are also tools to analyse SBF data files and convert them to various other formats [*RxTools User Manual* 2013].

The open source GNSS toolkit RTKlib is used for self-positioning with standard and high precision [Takasu 2013].

After recording, the following post-processing is used:

1. Tracks' saved .ubx files are transformed to RINEX 3.02 [International GNSS Service (IGS) and Maritime Services Special Committee 104 (RTCM-SC104) 2013] format using the RTKlib library `convbin` module.

2. The .SBF files from base and rover, saved to SD cards of Altus APS 3 antennas, are transformed into RINEX format using the `SBF converter` module from RxTools.

3. The rover's perception and reference, as well as pedestrian tracks along with the base RINEX files are used to calculate RTK mode for the rover and tracks using RTKlib's `rnx2rtkp` application. As soon as APS 3 has L1/L2 antennas, the RTK mode provides it with the fixed, highly precise position. For u-blox antennas with only L1 antennas, the precision is floating-point, but still RTK.

Another schema of RTK for Raspberries-Pi was envisioned. It is about online-correction sending directly to u-blox receivers. The corrections from Altus APS 3 base are sent via the TCP protocol to WiFi antennas, and the Linux socat utility retransmits them to u-blox. In that way, the receivers could output directly an RTK-corrected NMEA string. The method was rejected due to the latency on the reception of the base station correction, and since it does not offer post-processing flexibility.

According to the GPS technology, the rover is represented as a point in an LLA (Latitude, Longitude, Altitude) coordinate system. All the tracks are also represented as points in a LLA system. For further processing, some transformations between coordinate systems of different sensors, as well as between coordinate systems of GPS positioning, are carried out.

5.3 Coordinate systems

5.3.1 Objectives

Since the system has a set of sensors where each of them has its own coordinate system, one must know how to transform from one set of coordinates to another. Particularly, in this case, one can talk about four representation spaces:

1. The camera has two coordinate systems:
   — The 3D space system with origin at the current camera position.
   — The 2D space coordinate system which is a projection of the previous one to the recorded 2D image. It is discrete with pixel resolution and has its zero-point in upper left corner of the image.

   One can project points in 3D camera space onto the 2D projection using camera intrinsic parameters, but the inverse transformation is more difficult (impossible without some additional assumptions, like rover pitch equal zero everywhere and a ground modelled as a plane). Since the 2D camera projection can not be transformed to other coordinate systems, one must find a solution to project other coordinate systems to the camera one.
2. The LIDAR 3D space is analogous to camera 3D space: it is rover-fixed with its zero-point at the current LIDAR sensor position. It is possible to convert LIDAR 3D space to camera 3D space by a linear transformation which can be represented as rotation matrix and translation vector, since both spaces are Cartesian.
3. The GPS 3D space (latitude, longitude, altitude - LLA)is an Earth-centred global coordinate system, and to convert it to the rover ego-motion system one must apply several transformations including the estimation of the rover orientation.
4. The odometry space is a 2D Earth projection with a center in the initial rover position. Axes are defined by initial rover orientation. In the dataset, this coordinate system is not connected to the other ones and only given as a supplementary perceptual data.

A transformation schema between coordinate spaces is illustrated in Fig. 5.8.

5.3.2 Vision calibration

The base method is derived from Zhang's calibration technique [Zhang 1999] and needs several images of a chessboard captured from a camera under different angles. To project an arbitrarily oriented plane from world (chessboard) coordinate system to the 2S image projection, the following transformation is used:

$$s\mathbf{m} = \mathbf{H}M \tag{5.4}$$

with $\mathbf{H} = \mathbf{A}[\mathbf{r}_1\mathbf{r}_2\mathbf{r}_3\mathbf{t}] = \mathbf{A}[\mathbf{R}\mathbf{t}]$. Here matrix $\mathbf{A}$ is still the matrix of intrinsic parameters defined in Eq. 5.3, and $\mathbf{R}$ and $\mathbf{t}$ are extrinsic parameters which transform points from the world coordinate system to the camera's one. M and $\mathbf{m}$ are points in world coordinate system and 2D image respectively.

To estimate the intrinsic parameters, a maximum likelihood is obtained via minimizing the following expression:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\|\mathbf{m}_{ji}-\hat{\mathbf{m}}(\mathbf{A},\mathbf{R}_i,\mathbf{t}_i,M_j)\|^2 \tag{5.5}$$

where $\hat{\mathbf{m}}(\mathbf{A},\mathbf{R}_i,\mathbf{t}_i,M_j)$ is the projection of point M_j in image i, according to Eq. 5.4. The closed-form solution needs an initial guess of intrinsic matrix $\mathbf{A}$, $[\mathbf{R}_i,\mathbf{t}_i|i=1..n]$. The nonlinear minimization problem is solved using the Levenberg-Marquardt algorithm.

The radial distortion k_1 and k_2 is calculated either after solving the minimization problem by solving Eq. 5.6, or using their integration in the functional in Eq. 5.5.

$$\begin{bmatrix}(u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2\\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2\end{bmatrix}\begin{bmatrix}k_1\\k_2\end{bmatrix}=\begin{bmatrix}\hat{u}-u\\\hat{v}-v\end{bmatrix} \tag{5.6}$$

or, in matrix form $\mathbf{Dk}=\mathbf{d}$. Here x,y are normalized image coordinates of non-observable ideal images without distortion, $\hat{x}$,$\hat{y}$ are real distorted normalized image coordinates. Similarly, u,v and $\hat{u}$,$\hat{v}$ are ideal and corresponding real pixel coordinates. To eliminate distortion effects from the image using given distortion coefficients, Eq. 5.7 must be solved.

There is a strategy of alternating distortion coefficients and other parameters estimations until their Eq. 5.6 and Eq. 5.5 convergence [Zhang 1999].

$$\begin{aligned}\hat{x}&=x+x[k_1(x^2+y^2)+k_2(x^2+y^2)^2]\\ \hat{y}&=y+y[k_1(x^2+y^2)+k_2(x^2+y^2)^2]\end{aligned} \tag{5.7}$$

The calibration procedure was realized using the Camera Calibration Toolbox for Matlab [Bouguet 2003]. This tool helps to calculate the intrinsic matrix from Eq. 5.3 and distortion coefficients using 20-25 images of a planar checker-board registered in front of the camera on different angles, as it is shown on Fig. 5.6. The method is based on the maximum likelihood estimation of the Eq. 5.5.

The procedure of calibration consists of semi-automatically extracting image corners.

OpenCV is an open source computer vision tool [Bradski 2000]. OpenCV library's methods were used to undistort images. To eliminate black zones after undistortion and to remove the useless view if the rover's hood, images were cropped, and their final size is $1024x576$.

5.3.3 Vision-LIDAR extrinsic calibration

Using camera parameters, the method of video and LIDAR sensor alignment with circular target [Fremont, Sergio Alberto Rodriguez Florez, and Bonnifait 2012] was applied. This method determines the rotation and translation transformations relative to poses of sensors, while using sets of captured images and LIDAR points cloud corresponding circular targets for some various positions.

The calibration itself is based on the minimization of the functional in Eq. 5.8.

Figure 5.6 – Example frame of checker-board inner camera calibration

Figure 5.7 – Camera-LIDAR calibration with a circular target example frame

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{W}.\|\mathbf{p}_{ij}^{\mathcal{V}}-\mathbf{R}_{\mathcal{L}}^{\mathcal{V}}\mathbf{p}_{ij}^{\mathcal{LID}}-\mathbf{t}_{\mathcal{LID}}^{\mathcal{CAM}}\|^2 \tag{5.8}$$

here $\mathbf{p}_{ij}^{\mathcal{L}}$ is a jth point in LIDAR space for a ith circle pose, $\mathbf{p}_{ij}^{\mathcal{V}}$ is a point in camera space respectively. The $\mathbf{R}_{l}^{\mathcal{V}}$ and $\mathbf{t}_{\mathcal{L}}^{\mathcal{V}}$ are the rotation matrix and translation vector from LIDAR space to camera space, and the $\mathbf{W}$ is a weight matrix.

The circular target is detected by both camera and LIDAR. The LIDAR detects the inner edges of a circle, from which the calibration method reconstructs the circle's form. The camera's target representation is reconstructed using 16 points outlined manually from a 2D image projection. The camera's intrinsic parameters which are necessary for the correct camera space

positioning are taken from the calibration mechanism described in the previous section.

The method also provides a performance and an error estimation analysis. As a result, the linear transformation $\mathbf{T}_{\mathcal{L}}^{\mathcal{V}}$ between 3D LIDAR-centered space and 2D camera projection are obtained:

$$\mathbf{T}_{\mathcal{L}}^{\mathcal{V}} = \mathbf{A} \quad \begin{bmatrix} \mathbf{R}_{\mathcal{L}}^{\mathcal{V}} & | & \mathbf{t}_{\mathcal{L}}^{\mathcal{V}} \end{bmatrix} \tag{5.9}$$

where the matrix $\mathbf{A}$ represents the intrinsic camera parameters from Eq. 5.3.

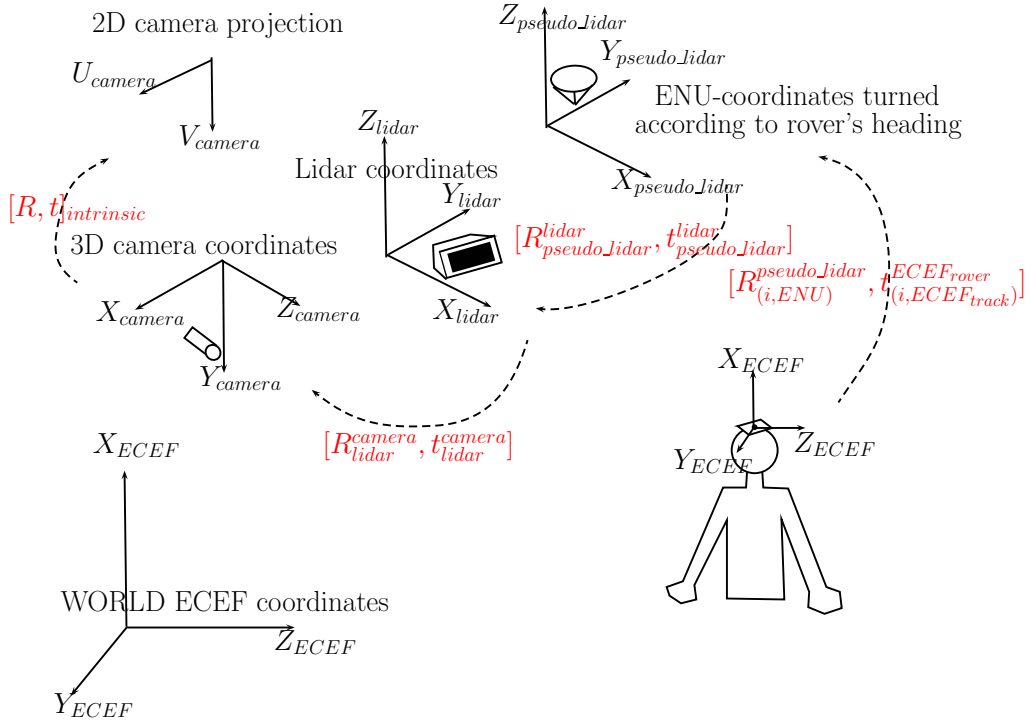


Figure 5.8 – Coordinate systems transformation schema

5.3.4 Vehicle-centered reference frame

The geographic coordinate system LLA is spherical and not adequate for road applications, because it provides distances in angles from the equator and the zero meridian, and not in meters on the Earth surface. Firstly, it must be transformed into ECEF ("Earth-Centered, Earth-Fixed") coordinates, which is Cartesian with zero point in the center of the Earth. The transformation is carried out as follows:

$$\begin{aligned} x &= (h + N)\cos\lambda\cos\phi \\ y &= (h + N)\cos\lambda\sin\phi \\ z &= (h + N(1 - E^2))\sin\lambda \end{aligned} \tag{5.10}$$

where $N = \frac{a}{\sqrt{1-E^2\sin^2\lambda}}$ is the Geoid radius at a given point, $a = 6378137.0$ is the equatorial radius, h is the altitude, λ and ϕ are latitude and longitude expressed in radians, $E = 0.081819191$ is the eccentricity of the Earth's elliptical cross-section. The transformation is based on the WGS 84 Coordinate System as on a Conventional Terrestrial Reference System [*World Geodetic System — 1984 (WGS - 84) Manual* 2002].

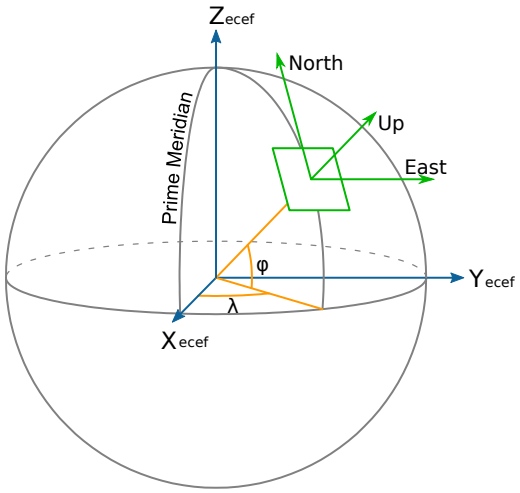


Figure 5.9 – Earth coordinate systems: LLA (ϕ,λ), ECEF, ENU (Eash,North,Up)

The ECEF coordinates can be transformed into ENU (East-North-Up) also known as a LTP (local tangent plane) coordinate system which give a ground orientation to east, north and vertical directions. It is important to know that ECEF to ENU conversion must always have a reference point denoted, the ECEF point that are transformed to zero center in ENU coordinates.

To transform ECEF coordinates to ENU, one must have also the LLA coordinates:

$$\begin{aligned} e &= -x\sin\phi + y\cos\phi \\ n &= -x\cos\phi\sin\lambda - y\sin\phi\sin\lambda + z\cos\lambda \\ u &= x\cos\phi\cos\lambda + y\sin\phi\cos\lambda + z\sin\lambda \end{aligned} \tag{5.11}$$

where x,y,z are ECEF coordinates in meters.

To transform LLA point to ENU coordinate with given reference zero point, one must subtract ECEF coordinates of reference point before conversion to ENU:

$$\begin{aligned} x &= x - x_{ref} \\ y &= y - y_{ref} \\ z &= z - z_{ref} \end{aligned} \tag{5.12}$$

That is because ECEF to ENU projection projects not the point itself, but the difference between reference point and given one. The difference in ECEF coordinates is a vector on an Earth surface, when the point itself is a vector from the Earth center to a point on its surface.

Let the LLA to ECEF transform be $\mathbf{T}^{\mathcal{ENU}_i}_{\mathcal{LLA}}$, let the ECEF to ECEF transform of given reference $\mathcal{ECEF}_i$ (as soon as the rover-centred coordinate system is needed) be $\mathbf{T}^{\mathcal{ECEF}_i}_{\mathcal{ECEF}}$ and let the ECEF to ENU transform be $\mathbf{T}^{\mathcal{ENU}_i}_{\mathcal{ECEF}_i}$. Then, the the LLA coordinates can be placed in ego-centred ENU coordinates of the rover using Eq. 5.13, where $\mathcal{ECEF}_i$ is a ECEF space with a reference of the actual rover position.

The $\mathcal{ENU}_i$ spaces are shown in Fig. 5.10 in comparison to the $\mathcal{ENU}$ space transformed from ECEF with only one reference point.

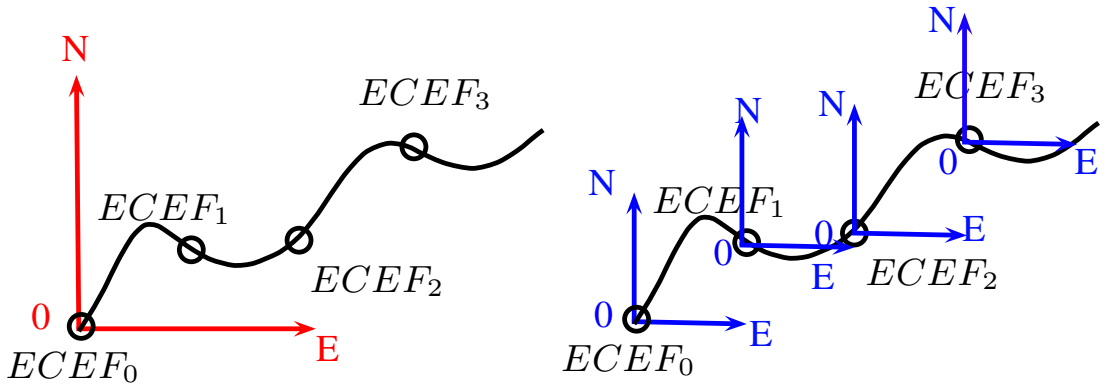


Figure 5.10 – Difference between ENU coordinates transformed from ECEF with one reference point or with references given from all ECEF points in series during rover motion

$$\mathbf{T}_{\mathcal{LLA}}^{\mathcal{ENU}_i} = \mathbf{T}_{\mathcal{ECEF}_i}^{\mathcal{ENU}_i}\mathbf{T}_{\mathcal{ECEF}}^{\mathcal{ECEF}_i}\mathbf{T}_{\mathcal{LLA}}^{\mathcal{ECEF}} \tag{5.13}$$

At the same time the rover ENU coordinates, with initial rover position as a reference point, are calculated using $\mathbf{T}_{\mathcal{LLA}}^{\mathcal{ENU}_0}$ of Eq. 5.14.

$$\mathbf{T}_{\mathcal{LLA}}^{\mathcal{ENU}_0} = \mathbf{T}_{\mathcal{ECEF}_0}^{\mathcal{ENU}_0}\mathbf{T}_{\mathcal{ECEF}}^{\mathcal{ECEF}_0}\mathbf{T}_{\mathcal{LLA}}^{\mathcal{ECEF}} \tag{5.14}$$

After getting localized objects in rover-centred ENU, one needs to turn axes Est-North to make them congruent with the axes of movement direction. To do this, the transformations in Eq. 5.15 and Eq. 5.16 are applied.

$$\mathbf{T}_{\mathcal{ENU}_i}^{\overline{\mathcal{L}}} = \begin{bmatrix} & \mathbf{R}_{\mathcal{ENU}_i}^{\overline{\mathcal{L}}} & \end{bmatrix} \tag{5.15}$$

$$\mathbf{R}_{\mathcal{ENU}_i}^{\overline{\mathcal{L}}} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.16}$$

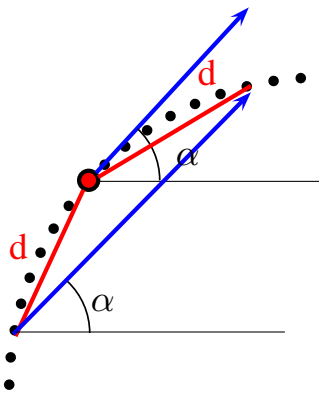


Figure 5.11 – Yaw estimation from curve position. When taking neighbouring points at a fixed distance d, problems with static or slow rover are avoided

Here, $\mathbf{R}_{\mathcal{ENU}_i}^{\overline{\mathcal{L}}}$ is a rotation matrix, α is an yaw angle between rover direction and the East calculated from rover neighbour positions in $\mathcal{ENU}_0$. One approximates α by an angle, connect-

ing the nearest future point being distant from the observed one at a fixed parameter value d, and the nearest past point being distant from the observed one at the same fixed value d, as it is shown on Fig. 5.11.

The resulted $\overline{\mathcal{L}}$ coordinates are already very close to the LIDAR's detections.

The axes of LIDAR are not aligned with the rover's pitch in ego-motion coordinates. To align them, a Singular Value Decomposition (SVD) method is applied on set of pairs $[p_{\overline{\mathcal{L}}}, c_{\mathcal{L}}]$, where $p_{\overline{\mathcal{L}}}$ is a GPS position transformed to pseudo-LIDAR coordinate system and $c_{\mathcal{L}}$ is a center of point cloud of an associated pedestrian. SVD estimates the rotation matrix to apply to the axes of $\overline{\mathcal{L}}$. Let the matrix $\mathbf{P}$ represent the input $N \times K$ data from $\overline{\mathcal{L}}$ space, where N is a number of points, and K is a point dimension. Let the $\mathbf{C}$ be an anologous data matrix $N \times K$ of points in the space $\mathcal{L}$. To estimate the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ which respect the transformation:

$$\mathbf{C} = \mathbf{R}\mathbf{P} + \mathbf{t}, \tag{5.17}$$

first the mean values are subtracted from coordinates:

$$\begin{aligned} \hat{\mathbf{P}} &= \mathbf{P} - \mathbb{E}[\mathbf{P}] \\ \hat{\mathbf{C}} &= \mathbf{C} - \mathbb{E}[\mathbf{C}] \end{aligned} \tag{5.18}$$

where $\mathbb{E}[\cdot]$ is expectation operator.

Then, the covaiance matrix $\mathbf{\Sigma}$ is calculated:

$$\mathbf{\Sigma} = cov(\hat{\mathbf{P}}^T, \hat{\mathbf{C}}^T) = \mathbb{E}[\hat{\mathbf{P}}^T \hat{\mathbf{C}}] \tag{5.19}$$

Now the SVD is applied: the covariance matrix is decomposed into three parts: two rotation matrices $\mathbf{U}$, $\mathbf{V}$ composed from left and right singular vectors, and one diagonal matrix $\mathbf{S}$ composed from singular values.

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{S}\mathbf{V}^* \tag{5.20}$$

where $\mathbf{V}^*$ is a conjugate transposed version of $\mathbf{V}$.

As soon as the one looks for a rotation matrix, the only matrices from singular vectors representing a rotational transform are taken into account:

$$\mathbf{R} = \mathbf{V}^{*\mathbf{T}}\mathbf{U}^{\mathbf{T}} \tag{5.21}$$

The translation vector is calculated from mean values:

$$\mathbf{t} = -\mathbf{R}\,\mathbb{E}[\mathbf{P}]^T + \mathbb{E}[\mathbf{C}]^T \tag{5.22}$$

As a result of imprecise heading estimation, $\overline{\mathcal{L}}$ and $\mathcal{L}$ has still a significant angle deviation when the rover turns. To eliminate this imprecision a mechanism of axes adaptation using SVD transformation is introduced. For each GPS detection in $\overline{\mathcal{L}}$ space, its own transformation to LIDAR space $\mathcal{L}$ is calculated according to Alg. 3. When LIDAR detections corresponding to the given GPS one are present, the SVD alignes GPS detection and LIDAR detection using a sliding window around the given pair LIDAR-GPS. If LIDAR detections are absent, the

alignment uses the nearest present pairs $[p_{\overline{\mathcal{L}}}, c_{\mathcal{L}}]$. Using this type of transformation, the GPS detections can be transformed finally to image 2D projection and serve as Ground Truth for vision. In Fig. 5.12, the initial 2D pair of axes $(10,0),(0,10)$ for the space $\overline{\mathcal{L}}$ are shown, as well as the axes transformed to the space $\mathcal{L}$ for all GPS detections of one scenario.

Algorithm 3: Alignment: *Pseudo LIDAR ($\overline{\mathcal{L}}$) and LIDAR ($\mathcal{L}$) coordinate systems alignment using SVD method*

Data: Set of LIDAR pairs $\mathcal{L}$ $[c_{\mathcal{L}}, t_{\mathcal{L}}]_i, i = 0..n_{objects}$ where $c_{\mathcal{L}}$ is a point from $\mathcal{L}$, $t_{\mathcal{L}}$ is a timestamp associated to this point for i object, whose number is $n_{objects}$. Set of GPS pairs $[p_{\overline{\mathcal{L}}}, t_{\overline{\mathcal{L}}}]_j, j = 0..n_{objects}$ in $\overline{\mathcal{L}}$. Set of rover reference timestamps t_r

Result: Set of rover yaw correction angles α_{corr} and 2D translation vector x_{corr}, y_{corr} to modify GPS detections

Interpolate LIDAR and GPS object's detections to timestamp t_r. Now there are $[p^{t_r}_{\overline{\mathcal{L}}}, t_r]$ and $[c^{t_r}_{\mathcal{L}}, t_r]$;

Set constants k_1 and k_2 - they signify the neighbourhood around given point where one looks for points for SVD ;

for *each* t_r **do**

Initialize SVD_{pairs} as empty pair structure;

if $p^{t_r}_{\overline{\mathcal{L}}}$ *exist* **then**

if $c^{t_r}_{\mathcal{L}}$ *exist* **then**

Let t_{r_k} be the the moment t_r plus k position in a set $[t_r]$;

k=0;

while $p^{t_{r_k}}_{\overline{\mathcal{L}}}$ *and* $c^{t_{r_k}}_{\mathcal{L}}$ *exist and* $k < k_1$ **do**

Add $[p^{t_{r_k}}_{\overline{\mathcal{L}}}, c^{t_{r_k}}_{\mathcal{L}}]$ to SVD_{pairs};

k=k+1;

k=0;

while $p^{t_{r_k}}_{\overline{\mathcal{L}}}$ *and* $c^{t_{r_k}}_{\mathcal{L}}$ *exist and* $k > -k_1$ **do**

Add $[p^{t_{r_k}}_{\overline{\mathcal{L}}}, c^{t_{r_k}}_{\mathcal{L}}]$ to SVD_{pairs};

k=k-1;

Apply SVD method for SVD_{pairs} and get $\alpha_{corr}, x_{corr}, y_{corr}$ for the moment t_r

else

Find nearest existed pair $p^{t_{r_k}}_{\overline{\mathcal{L}}}, c^{t_{r_k}}_{\mathcal{L}}$ for k>0;

while $p^{t_{r_k}}_{\overline{\mathcal{L}}}$ *and* $c^{t_{r_k}}_{\mathcal{L}}$ *exist and* $k < k_2$ **do**

Add $[p^{t_{r_k}}_{\overline{\mathcal{L}}}, c^{t_{r_k}}_{\mathcal{L}}]$ to SVD_{pairs};

k=k+1;

Find nearest existed pair $p^{t_{r_k}}_{\overline{\mathcal{L}}}, c^{t_{r_k}}_{\mathcal{L}}$ for k<0;

while $p^{t_{r_k}}_{\overline{\mathcal{L}}}$ *and* $c^{t_{r_k}}_{\mathcal{L}}$ *exist and* $k > -k_2$ **do**

Add $[p^{t_{r_k}}_{\overline{\mathcal{L}}}, c^{t_{r_k}}_{\mathcal{L}}]$ to SVD_{pairs};

k=k-1;

Apply SVD method for SVD_{pairs} and get $\alpha_{corr}, x_{corr}, y_{corr}$ for the moment t_r

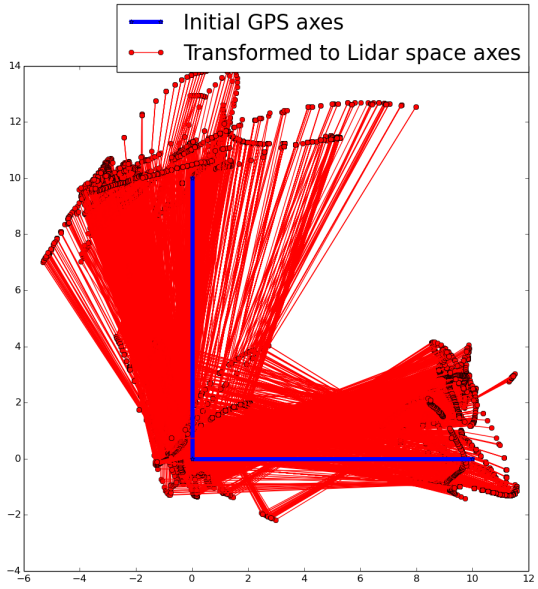


Figure 5.12 – $\overline{\mathcal{L}}$ axes (blue) and its SVD transformations to $\mathcal{L}$ (red) for all GPS detections of one scenario. The axes are shown as vectors (10,0), (0,10). The images show the rotation and translation transformations from $\overline{\mathcal{L}}$ to $\mathcal{L}$ for each timestep of the scenario

5.4 Files

The dataset has a file representation. Each recorded scenario has:

1. A file providing real-time perception data
2. A set of files with raw GPS data
3. A set of files with perception data resulting from offline processing
4. A file with Ground Truth annotations
5. Calibration data

In this section there is a full documentation about all these types of data.

5.4.1 ROS file format

The Robot Operating System (ROS) is a set of software libraries and tools that help to build and manage complex robot applications [Quigley et al. 2009]. For the dataset application, ROS is needed to provide data bags (so called ROSbags), which are capable to store signals from all sensors in real time and to replay them later (also in real time). ROS also provides a set of drivers to parse and encapsulate messages from devices. The common schema of ROS is shown in Fig. 5.13. All messages are published in "topics" with a given queue size. When some process (a "node") subscribes to a topic, the appearance of a message in this topic activates a processing mechanism inside the node. When new message arrives into a full queue, the queue removes the oldest message. That system allows to simulate and work with real-life multi-thread messages easily. That is why ROS was chosen as a real-life version of the dataset. The offline version as a set of text files and images is proposed too.

Almost all sensors' detections are transformed into ROS messages. For camera's images, a caption driver FlyCapture SDK is used with modifications allowing to transform images to

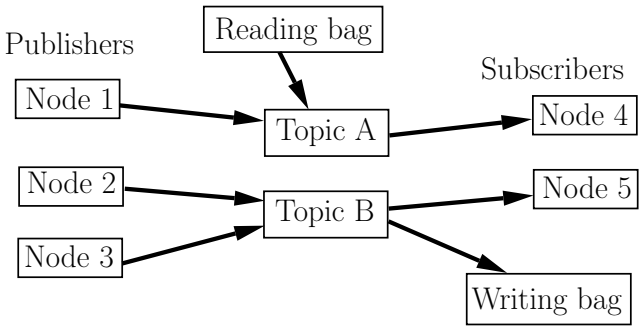


Figure 5.13 – Base concept of ROS: nodes make messages, publish them in topics. Other nodes can subscribe to topics and read published messages

ROS messages of type `sensor_msgs::Image` illustrated with details on List. A.3. The LIDAR detections are processed with a `sick_ldmrs` ROS package with its executable script `sickldmrs.py` [Grandbois and Pauling 2012] to compose and send ROS messages of type `sensor_msgs::PointCloud2` explained in List. A.4.

Wheel encoders are connected to rover's bus CAN (Controller Area Network) and transmit messages encoded in $TPCANMsg$ format of *libpcan* library:

1. DWORD ID - id of the sender encoder
2. BYTE LEN - length of actual message in bytes. Here it is not used, as soon as the length is only one and is known
3. BYTE MSGTYPE - type of used message. Here it is not used
4. BYTE DATA [8] - all the rest of essencial encoders information:
   — 1-4 bytes - number of impulses read from sensors. This value shows absolute wheel position changing
   — 5-6 bytes - number of impulses changing, i.e. wheel speed in terms of impulses
   — 7-8 bytes - timestamp

Each of encoder send message with 10 ms frequency. CAN bus is parametrized to have 1 Mb/s rate. Bus CAN is connected to a PC with USB adapter and its messages are stored timestamped with the POSIX time of the PC.

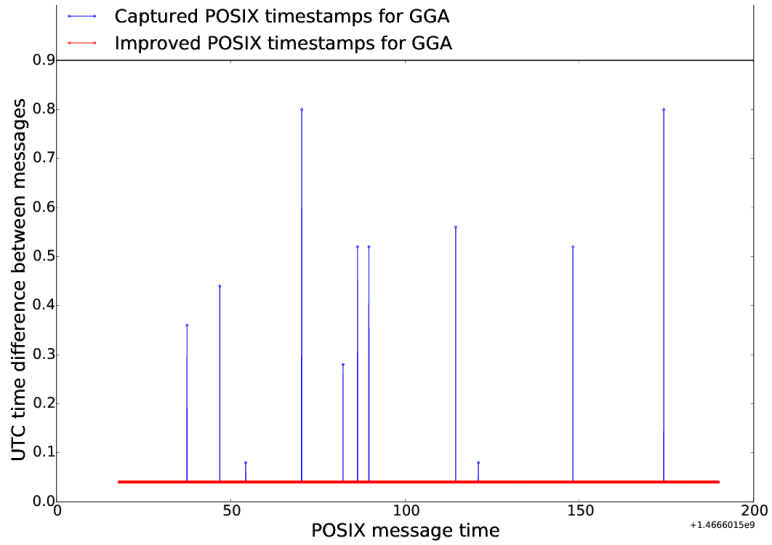
A `peak-linux-driver` based ROS package transform CAN output to ROS message of new defined type `pcan_msgs::CAN` shown in List. A.1. Online NMEA output from the rover's receivers is transformed into ROS messages of type `std_msgs::Header` as explained in List. A.2.

5.4.2 Time alignment

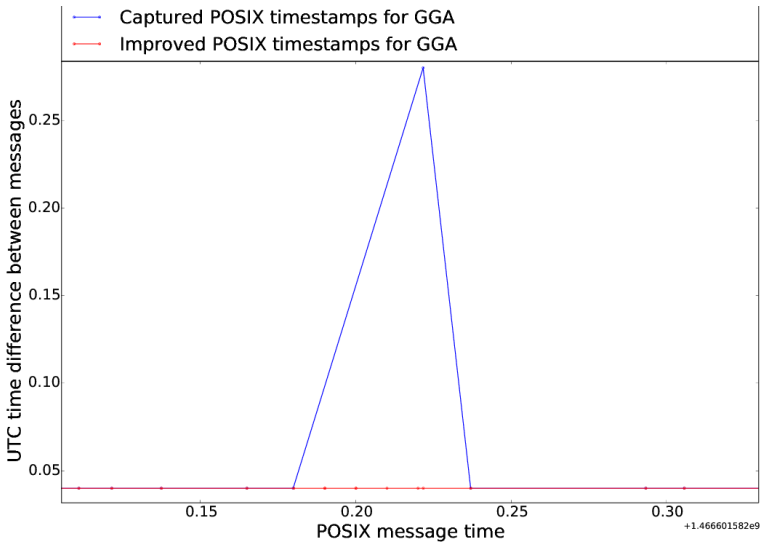
All types of used ROS drivers make timestamps for each data structure using host PC's system clock in POSIX format. For the LIDAR, one sweep is a $frame$ and its timestamp is a time of ROS-message creation. For a camera $frame$, its time-stamp is also a time of ROS-message creation. The same principle is valid for odometry messages. GPS files are saved in offline-mode, they contain only UTC time provided by satellites with resolution which is not high enough to be matched to the local POSIX time of the host PC.

As it was mentioned above, two GPS receivers, one U-blox and one APS 3, installed on the rover's top are configured to output NMEA string via serial connection to the PC processing

all other sensors' messages. This is needed to associate the PC's POSIX time with GPS timestamps and to thus have all sensor readings time-stamped. Each NMEA message that arrives via the serial port is processed by the ROS driver making messages of types `std_msgs::Header`, containing NMEA string and the POSIX timestamp of the incoming message. Most GPS timestamps have a corresponding POSIX timestamp. In case of lost NMEA messages, a procedure of reconstruction of missing timestamps is proposed: the area of lost timestamps is filled with fixed time steps, as it is shown in Fig. 5.14b.



(a) Reference rover time alignment for an example scenario



(b) One time hole filling example

Figure 5.14 – The graphs show POSIX timestamps (horizontal axis) and time difference between neighbouring timestamps (vertical axis). The blue line signifies original, not improved POSIX timestamps for GGA messages. One can see some high differences between neighbouring timestamps, indicating loss of time information. The red line shows the improvement by filling "lost" timestamps

5.4.3 Extracted files

Besides the online-format ROS files, also more classical offline-processing-style files are provided. As it is shown on Fig. B.1, the visual perception data is represented as a set of *.png*

images and a *.txt* file containing timestamps for them. The range perception data is represented as a set of *.txt* files for each scan. The timestamps are stored in a similar manner in a *.txt* file. GPS data is stored in text files as NMEA string lines with their POSIX timestamps. For each track, one file is provided as well as for the GPS perception and reference data of the rover. Encoders' messages for the odometry reconstruction are given in their raw form to allow for methods of optimal wheel radius and wheelbase estimation.

5.4.4 GPS raw files

The raw GPS offline data is provided too. It represents APS-3 logged *.sbf* files for the base and reference rover. For U-blox receivers, raw *.ubx* files are also present. Text files of *.pos* extensions are NMEA-form output resulting offline manipulations with GPS raw data. Theoretically, they can be recalculated using other extraction methods as described in Sec. 5.2.2, or using other parameters.

5.4.5 Transforms files

A set of files with transformations between sensors spaces are provided too:

— Intrinsic parameters of the camera, calibrated with methods described in 5.3.2 are completed with the raw images of a checker-board under different angles.

— LIDAR-camera transform detailed in 5.3.3 is provided with raw images circular target filmed under different angles with the corresponding LIDAR's range point clouds

— GPS-LIDAR fitting transforms mentioned in 5.3.4 are proposed too.

5.4.6 Annotations

The main goal of the dataset is to provide a complete experimental tool to test various approaches in multi-object tracking and detection in multi-modal perception. To this end, Ground Truth information is needed. Ground Truth (GT) is the ideal result expected after the application of tracking algorithms. For the case of tracking, GT is a trajectory of each dynamic object in the field of view of the sensors. In case of GPS, Ground Truth is a trajectory composed from 3D points in the geographical coordinate system, where for each time-stamp an object is expressed as only one point. In case of LIDAR, Ground Truth for one time-stamp is a cluster of 3D points in the LIDAR-centred Cartesian coordinate system, which are points reflected by the objects while scanning. In case of camera-based visual information, a GT object at a time-stamped 2D rectangle, circumscribing the visual contour of an object. All these types of GT are illustrated in Fig. 5.15.

In this dataset, the Ground Truth is given in the form of *.xml* files of the type illustrated in List.C.1.

The GPS type of Ground Truth is given as GPS fixes without special GT extraction. The LIDAR GT is extracted with the semi-automatic procedure based on expert (human) track initialization and automatic point cloud tracking. The vision GT is based on GPS projection onto a 2D image as described in Sec. 5.3.4, with human's height assumed to be roughly equal to

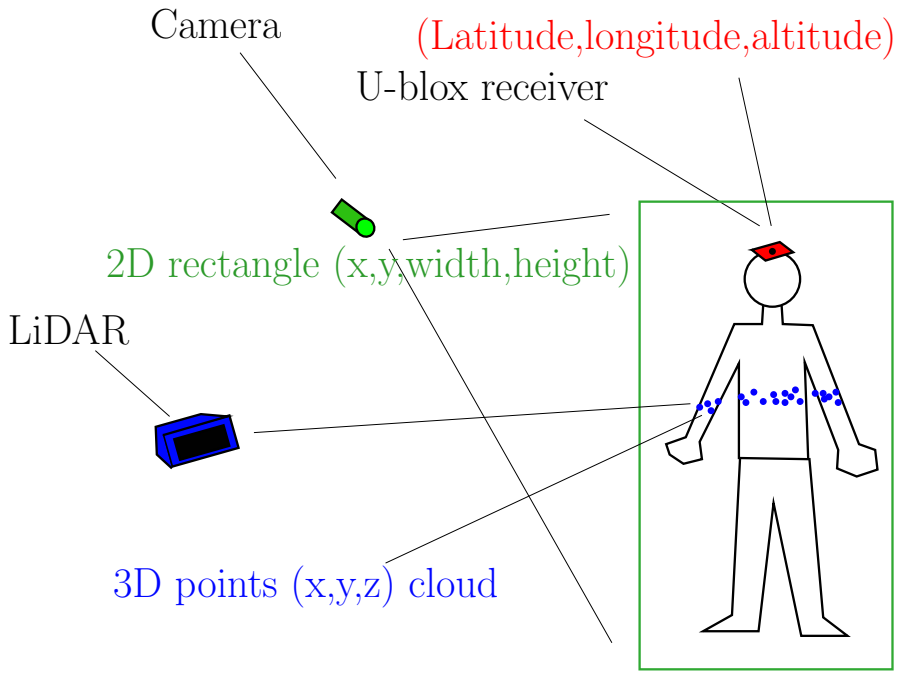


Figure 5.15 – Ground Truth representation for various sensors: camera (image) GT is a 2D rectangle, LIDAR GT is a cloud of 3D points, GPS GT is one point

the rover's top and a human's width assumed to be 1 meter. The developpment kit is written in C++ which helps to generate Ground Truth files and is shown in Fig. 5.16.

Another tool of the development kit in order to verify the integrity of dataset is proposed. It is written in Python and represents a set of scripts, including a file `dataLoader.py` containing structures representing GT tracks and the functions to load them from *.xml* files. Another script `encoders_reader.py` reconstructs the odometry trajectory from raw encoder messages. The script `tracklets_reader.py` visualizes Ground Truth tracks and projects detections from one sensor space to another.

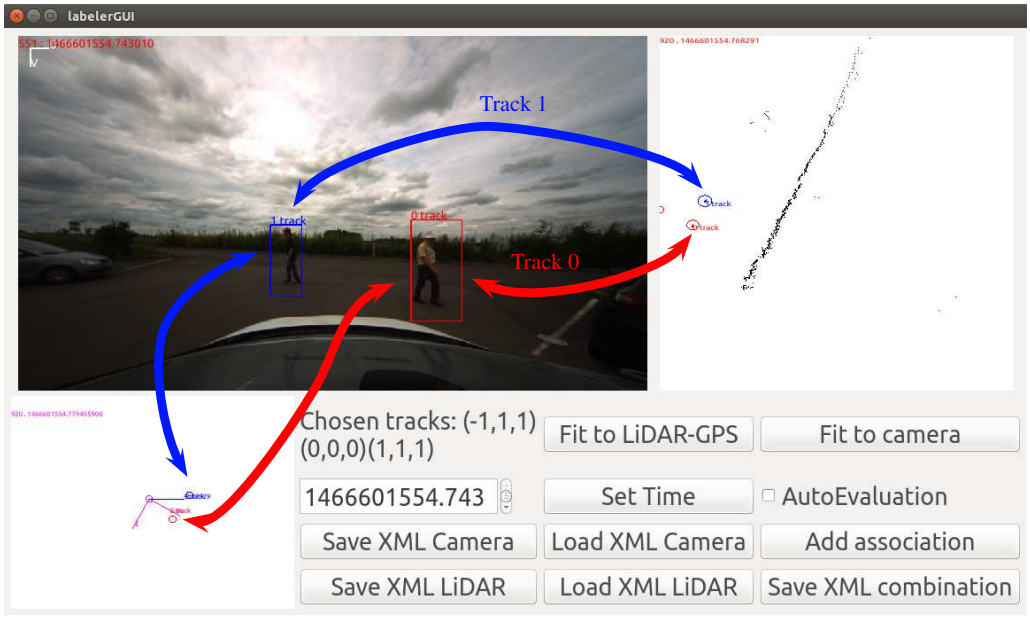


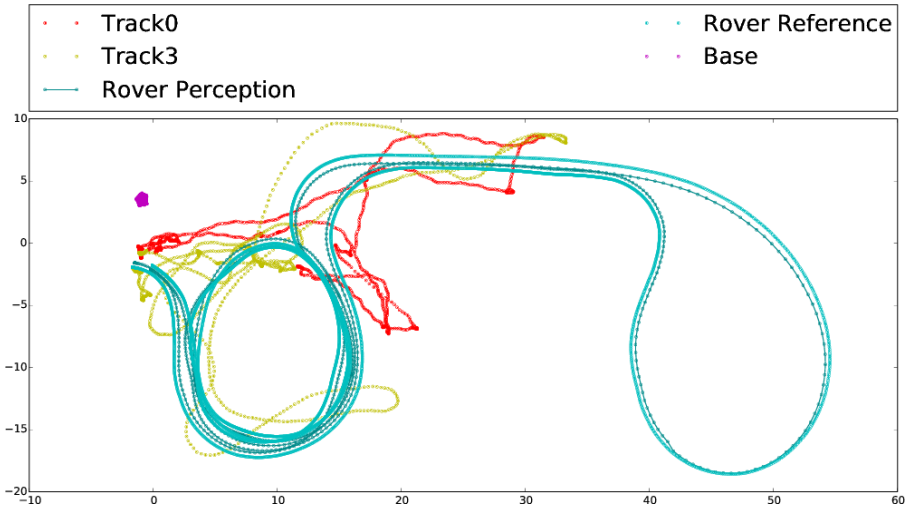
Figure 5.16 – Development kit for Ground Truth generation: in the top left corner, the visual annotations are shown, the top right corner contains the LIDAR scene with GT annotations (coloured circles are tracks). The bottom left corner shows GPS annotations in rover-centred Birds' Eye View. Arrows show associations between tracks.

5.5 Recorded scenarios

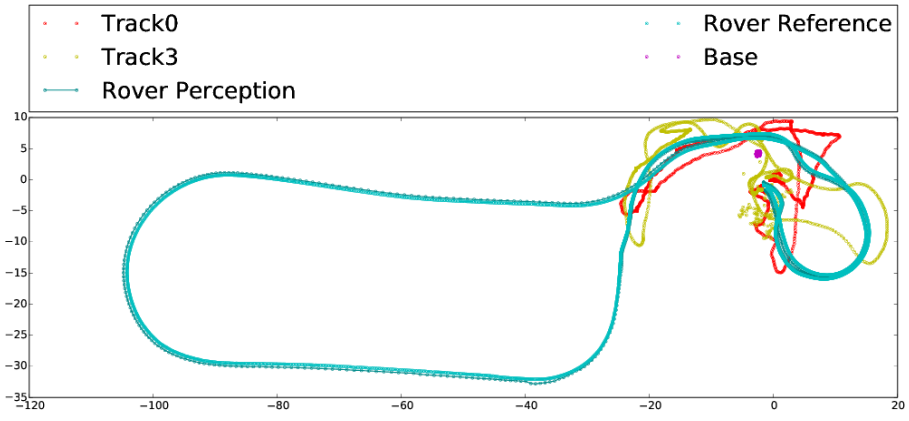
4 scenarios were so far recorded in different semi-urban environments, near to University Paris Saclay in France's Paris metropolitan region. They contain at most four tracked pedestrians moving around a slowly moving rover. The recorded scenarios have been defined for their difficulty w.r.t. sensor fusion and tracking, as evidenced by track intersections, occlusions, partial visibility etc. The descriptive chart with total Ground Truth detections number and duration is given in Tab. 5.3:

Scenario	Duration	GPS GT number	Video GT number	Range GT number
Scenario 1	2:51s	1719	2491	2316
Scenario 2	2:37s	1554	2800	3001
Scenario 3	3:03s	3672	9320	12016
Scenario 4	3:22s	4044	10045	14735

Table 5.3 – Scenarios descriptive chart



(a) Scenario 1



(b) Scenario 2

Figure 5.17 – Scenario's GPS trajectories for the base, perception, rover reference and tracks. The reference point is the first rover reference position. The base positions are processed in single, fixed mode, so the shown dispersion gives the order of corrections used in RTK mode for rover and tracks

Since the rover and pedestrians are equipped with GPS, one can plot the full scenario as

their coordinates on the Ground plane, as it is shown on Fig. 5.17a5.17b.

Also, one can try to measure the U-blox receiver's intrinsic precision. This precision is calculated as relative to the Altus APS-3 receiver position, which is supposed to be centimeter-precise. The rover perception is compared to the rover reference position. The horizontal mean distance between them is

$$m_h = \frac{1}{N}\sum(\sqrt{(e_p - e_r)^2 + (n_p - n_r)^2} \tag{5.23}$$

where N is a number of found pairs of perception and reference, e_p, e_r are East coordinates of perception receiver and reference receiver, n_p and n_r are the corresponding North coordinates. The vertical mean distance is measured as

$$m_v = \frac{1}{N}\sum abs(u_p - u_r) \tag{5.24}$$

where n_p and n_r are Vertical (Up) coordinates. The horizontal dispersion is calculated as

$$d_h = \frac{1}{N}\sum abs(m_h - \sqrt{(e_p - e_r)^2 + (n_p - n_r)^2}) \tag{5.25}$$

and vertical one as

$$d_v = \frac{1}{N}\sum abs(m_v - abs(u_p - u_r)) \tag{5.26}$$

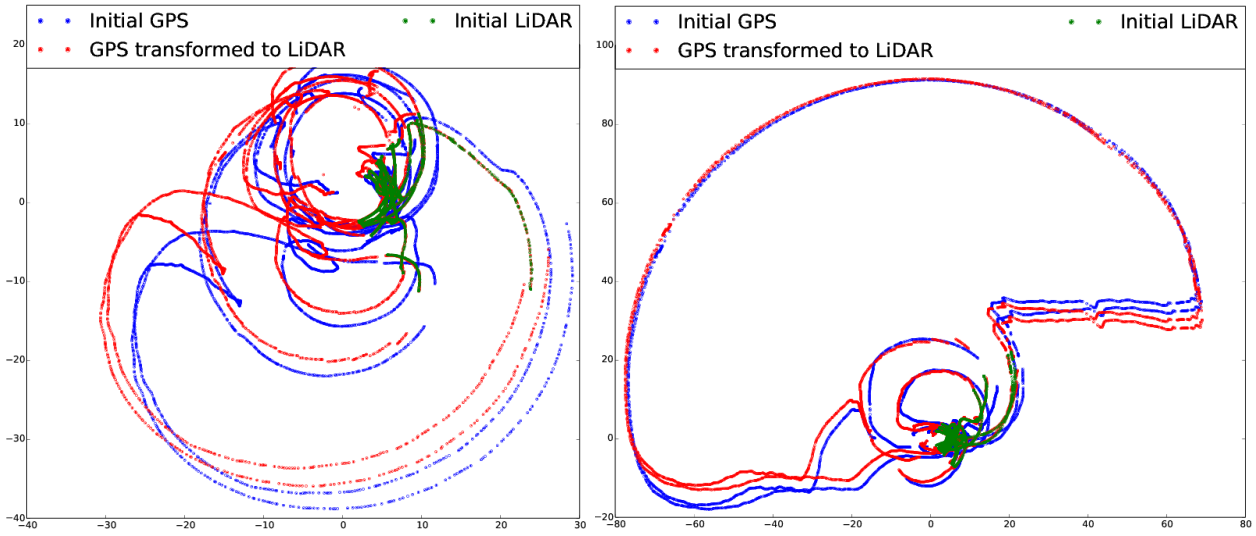
The calculated values are illustrated on the Tab. 5.4. The high value of vertical mean difference for the Scenario 2 is the consequence of the rover's trajectory near a high building. Some part of sky is occluded and the satellite data is less precise. In the same time, a post-processing filtering algorithm keeps the erroneous value constant, and the corresponding dispersion stays low. The horizontal values are much preciser: 40-41 cm is a real distance between receivers. Fluctuations of 6-8 cm in U-Blox positioning correspond to expectable values.

Scenario	Horizontal mean	Vertical mean	Horizontal dispersion	Vertical dispersion
Scenario 1	0.406919085187	0.113258635022	0.0660296133688	0.138379026754
Scenario 2	0.41329173529	1.53703328148	0.0825748695907	0.0734323213258
Scenario 3	0.419925052762	1.09849766286	0.211421303392	0.0960270613371
Scenario 4	0.540115957438	0.192691456112	0.236486342331	0.0113630018678

Table 5.4 – Measured perception of rover precision from the recorded scenarios in meters. The actual relative positions of rover receivers are not changed

The results of GPS detection corrections from pseudo-LIDAR space $\overline{\mathcal{L}}$ to $\mathcal{L}$ as they are descibed in Sec. 5.3.4 are shown in Fig. 5.18a5.18b. In these figures, the trajectories of GPS detections before and after corrections and also the LIDAR detections are traced. One can observe that after the matching, GPS is very coherent LIDAR detections trajectory, as it was expected.

Another result to characterize the quality of the recorded data is the odometry estimation. This is shown in Fig. 5.19a5.19b5.19c5.19d. They illustrate trajectories in the first two scenarios, as reconstructed from front or rear wheels. The coordinate system depends a lot on the initial



(a) Scenario 1 (b) Scenario 2

Figure 5.18 – Scenario's GPS transformation to LIDAR space. Initial GPS detections in pseudo-LIDAR space are shown, as well as LIDAR detections and GPS transformed to LIDAR space

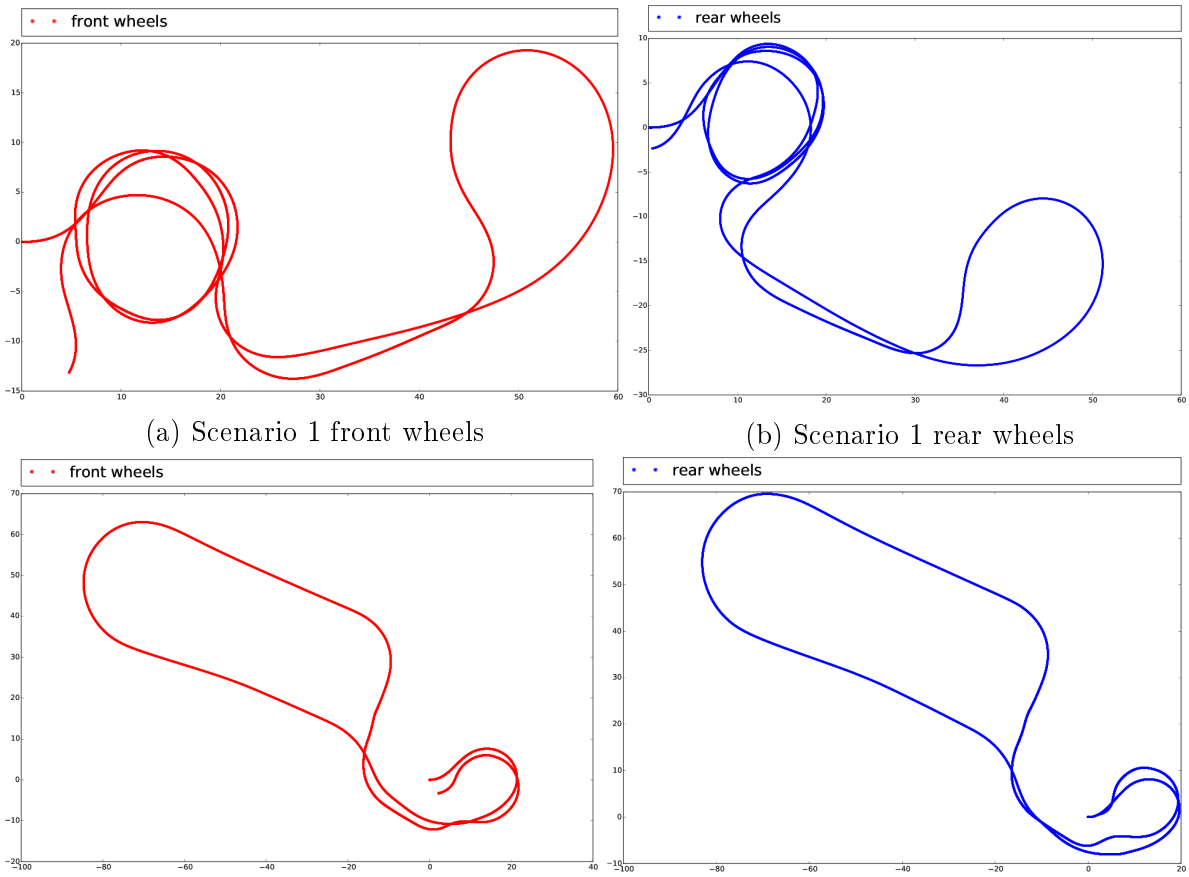
direction of movement, which is why for front and rear wheels of the first scenario the axes are rotated.

It is possible to measure the relative precision between odometry reconstructed trajectories by comparing with GPS reference trajectories. SVD transform is used to align ENU and wheel-based coordinate systems. Results are illustrated in Fig. 5.20, where WSS based reconstructed trajectories of Scenarios 1 and 2 are compared with GPS. Results were quantified computing the mean distance between corresponding points of two trajectories, see Tab. 5.5. Odometry errors are higher on Scenario 1 since the vehicle perform multiple turns. Such a kind of trajectories induces wheels sliding errors. The quantified error integrated on WSS trajectories confirms the integrity of the recorded data. This information is precise enough to carry out motion compensation on detected objects.

Scenario	Front wheels (m)	Rear wheels (m)
Scenario 1	1.9012	1.4271
Scenario 2	1.4122	1.2627
Scenario 3	1.9337	1.1880
Scenario 4	1.8389	1.2211

Table 5.5 – Odometry precision measurement chart. Columns correspond to mean distance between points of GPS and odometry reconstructed trajectories from rear and front wheels.

For now, the technical descriptions of the recorded scenarios is completed. Their semantic meanings with their applications for various evaluation purposes are described in the next chapter.



(a) Scenario 1 front wheels

(b) Scenario 1 rear wheels

(c) Scenario 2 front wheels

(d) Scenario 2 rear wheels

Figure 5.19 – Scenarios odometry reconstructed traces for front and rear wheel pairs

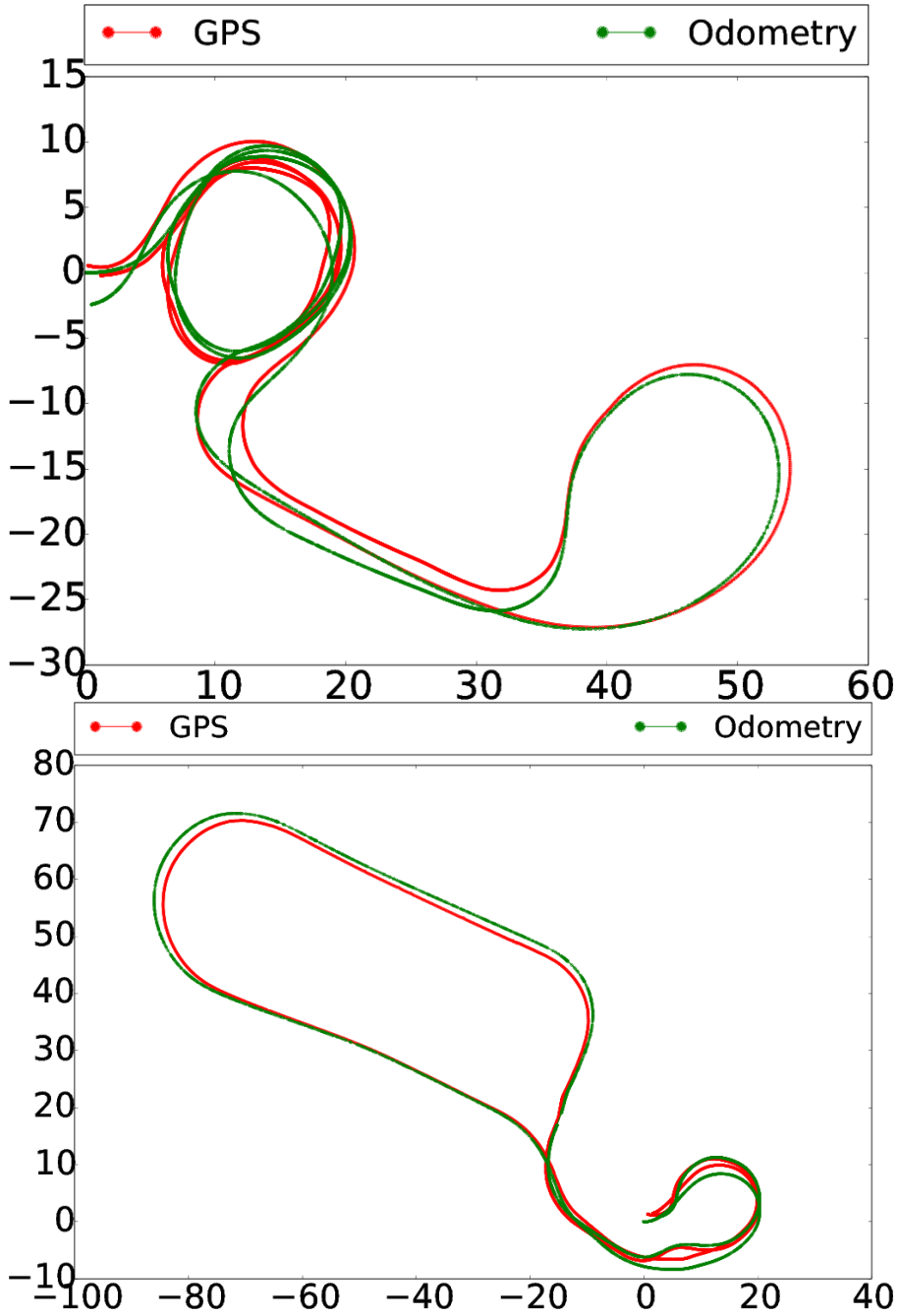


Figure 5.20 – Rear wheels-based odometry reconstructed traces matched with GPS trajectories with SVD transform. Scenarios 1 (left) and 2 (right) are presented.

Chapter 6

Experimental results

Contents

6.1 Introduction

Chap. 5 was intended to acquire data necessary to evaluate the efficiency of the proposed methods described in Chap. 2,3 and 4. Thus, the experiments reported in this chapter provide a full-scale proof of concept. The dataset is composed of four scenarios that were recorded with a specific purpose:

Scenarios 1 and 2 address classical and simple tracking situations. In the sequence, two pedestrians are moving in front of the rover. Pedestrians intersect the rover trajectory, follow one side of the road, or just stand quietly while the rover moves. Occasionally, the pedestrians are also recorded in a blind zone of the LIDAR, but still visible by camera. Since only few objects are present at each sampling time, this kind of scenarios are well suited as a training set for the data association method detailed in Chap. 3.

Scenario 3 aims at demonstrating the quality of the proposed tracking system. This scenario is composed of four pedestrians frequently occluded in a highly dynamic scene (e.g. round-about). This scenario provides a complex configuration for the association and tracking method.

Scenario 4 is intended to validate context-aided tracking concept. Four pedestrians are recorded: two of them evolve following an expected context motion and the others move following an unexpected context behaviour. This scenario provides a well-suited situation to identify two tracking groups.

6.2 Tracking

In the dataset, the Ground Truth (GT) is composed of LIDAR, ROI images and GPS annotations (i.e. tracklets). Object tracking algorithms can then perform taking advantage of on-board vehicle sensors only such monocular vision and LIDAR measurements (i.e. exteroceptive information). The GPS positioning of tracked objects is provided as a reference. Three evaluation criteria are investigated: continuity, overlap and Euclidean distance. In order to quantify these criteria, the proposed algorithms were evaluated under two different conditions: (1) Detections are emulated as all presented GT objects (2) Detections are emulated by the set composed of all presented GT objects and some false negatives. The objects positions are perturbed by an additive white noise.

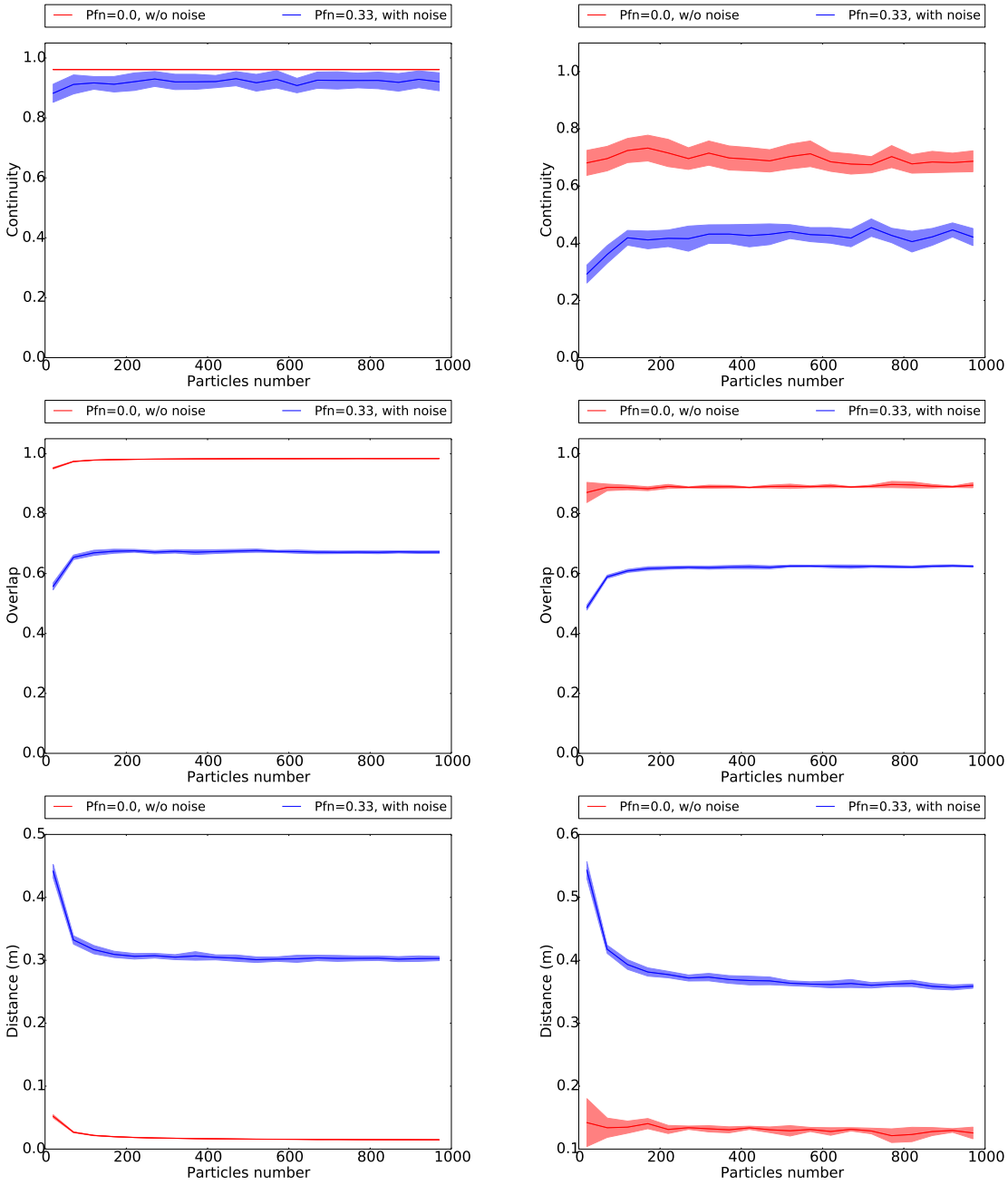


Figure 6.1 – Continuity, overlap and Euclidean distance for tracking in LIDAR range view of scenarios 1 (left column) and 3 (right column). 20 trials are carried out so as to achieve representative statistics. Two cases are evaluated: without noise in the detected object positions and with white noise in the detected object positions and false negative detections.

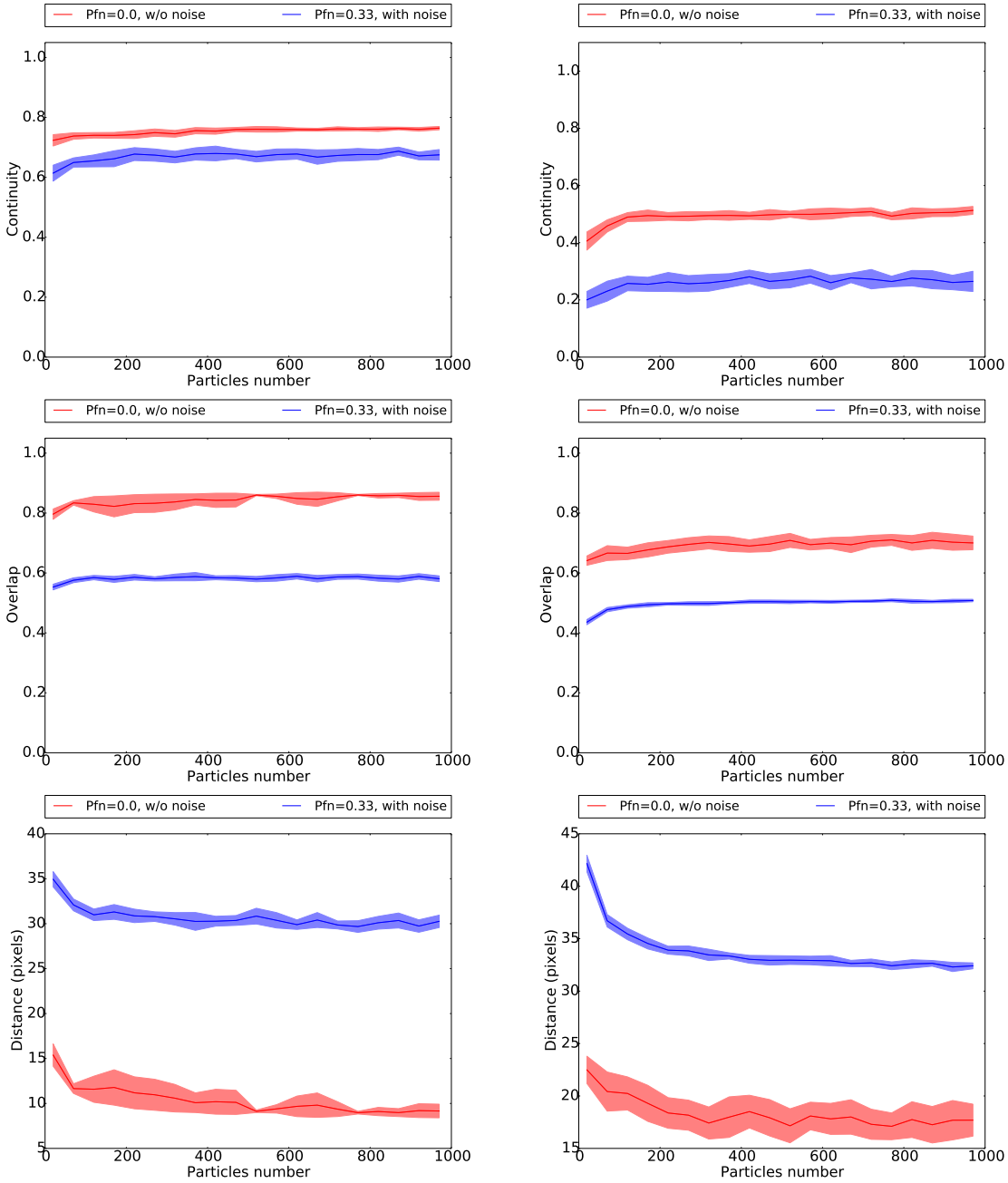


Figure 6.2 – Continuity, overlap and Euclidean distance for tracking in image projection view of scenarios 1 (left column) and 3 (right column). 20 trials are carry out as a representative statistics. Two cases are tested: without noise and with additive white noise and false negative detections.

In Fig. 6.1 and Fig. 6.2, the evaluation of two scenarios are illustrated: Scenario 1 is a low-speed simple scene and Scenario 3 is a more complex scene covering tracking issues like occlusions and turning trajectories.

For the object tracking algorithm performing on LIDAR measurements, the parameters were set: $R_b = 0.7$, $R_d = 0.1$ and $R_{ret} = 0.2$. Additive noise parameters applied to LIDAR observations: $\sigma_c = 0.3$ (m) and $P_{fn} = 0.33$.

The settings for object tracking performing on monocular vision: $R_b = 0.7$, $R_d = 0.1$ and $R_{ret} = 0.2$. Additive noise parameters applied to visual observations: $\sigma_c = 20$ (pixels) and $P_{fn} = 0.33$.

Based on Fig. 6.1 and Fig. 6.2, some insights are stated hereafter:

Tracking in LIDAR coordinate system is more stable and precise than on the image projec-

tive space. This was also noted on results are reported in Chap. 2, more particularly in Fig. 2.6 and in Fig. 2.7. It is worth noting that the choice of tracking parameters is not a trivial problem because of the perspective effects on the image plane. That is, depth changes not only lead to object size changes but also to different error distributions of the observed dynamics.

The increase of noise negatively impacts the continuity of tracks according to the chosen criterion. For KITTI dataset, in case of a high noise influence, the criteria overlap and Euclidean distance follow slight improvements. That is, a new track is initialized after a previous track loss. A new track perfectly matches to the corresponding detection. For the proposed dataset this effect is too week to be observed.

Finally, noise increases the uncertainty in tracking. Thus, this relation was observed in the dispersion of the computed criteria values.

6.3 Multi-modal association

The designed multi-sensor data association mechanism was evaluated on the proposed dataset following the same procedure as for KITTI and Honda datasets. In Fig. 6.3 is shown

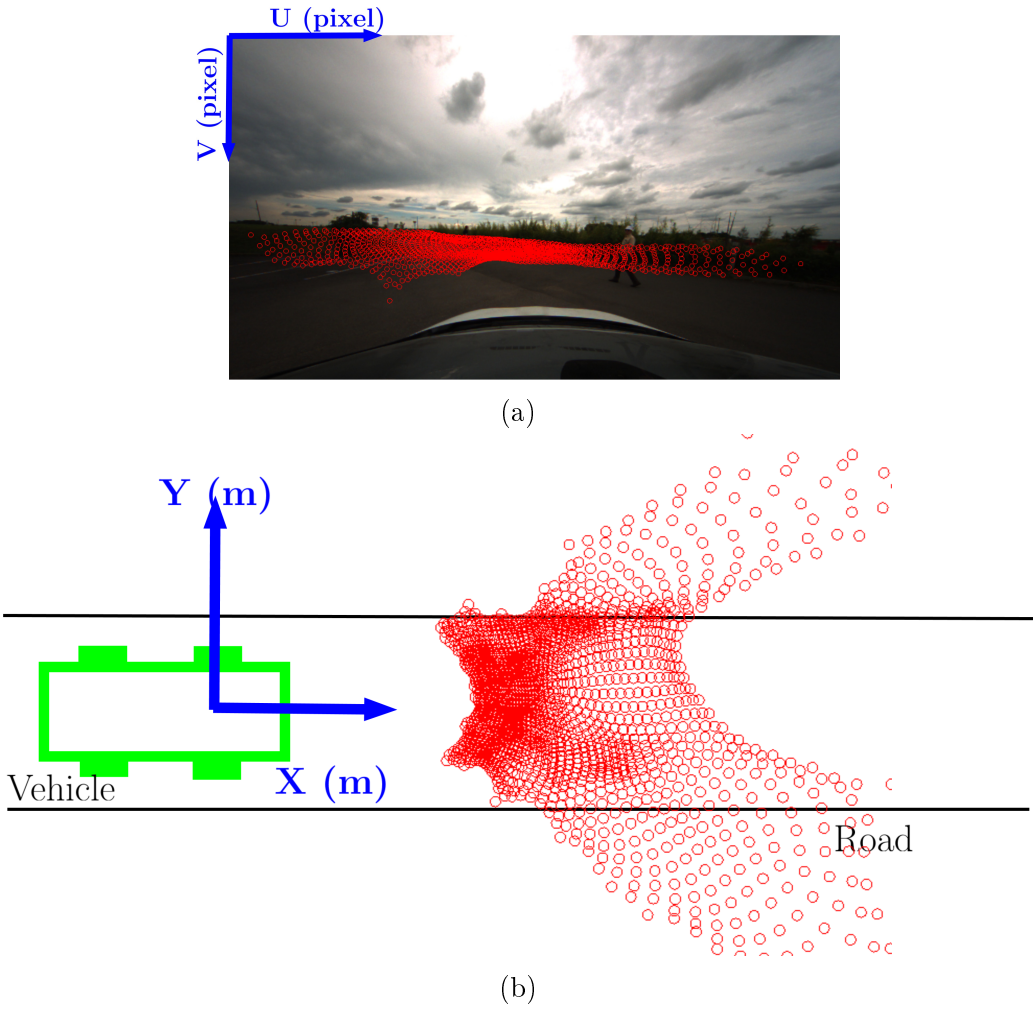


Figure 6.3 – Statistical models of sensory spaces are acquired by SOM for visual (a) and LIDAR sensors (b). Red points represent the position of SOM prototypes in the space of each sensor. The local density of prototypes is guided by average local density of data points. The data for SOMs constructions are GT annotations of the dataset proposed in Chap. 5

the resulting SOMs using GT detections from both, visual and LIDAR spaces.

In Fig. 6.4, the ROC curve for all recorded scenarios is illustrated. The training and the evaluation sets are composed from randomly chosen frames of a set containing all recorded scenarios. Because of the limited number of recorded scenarios, we consider necessary to apply also a cross-validation mechanism. In Fig. 6.5 ROCs graphs are provided for training-testing pairs, where training data are selected based on all recorded scenarios but one, this remaining set was devoted as the training scenario.

The analysis of Fig. 6.5, 3.8, 6.3 leads us to the following conclusions:

The SOM nodes distributions characterize the small data size: there are represented through areas with higher nodes densities. A similar trend was observed using Honda dataset (see Fig. 3.4d and Fig. 3.4c). On the contrary, in the KITTI dataset nodes are distributed more uniformly (see Fig. 3.4b and Fig. 3.4a).

The ROC curves corresponding to the proposed dataset are as smooth as ROCs for KITTI

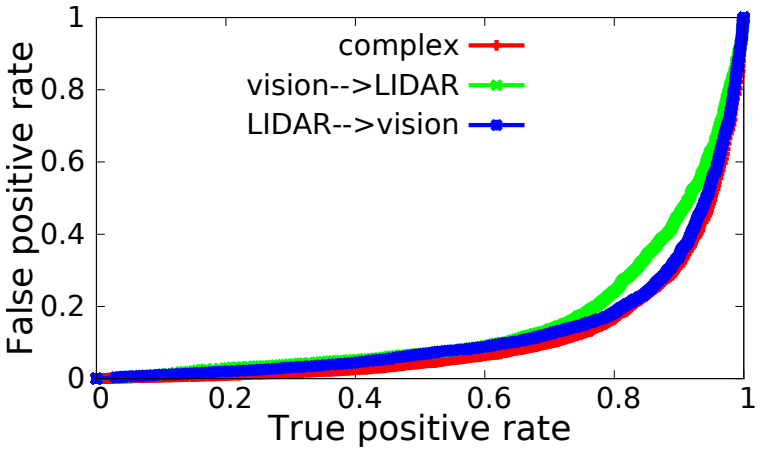


Figure 6.4 – ROCs for vision–>LIDAR, LIDAR–>vision and complex cross-verified strategy of Eq. 3.8 applied on proposed dataset.

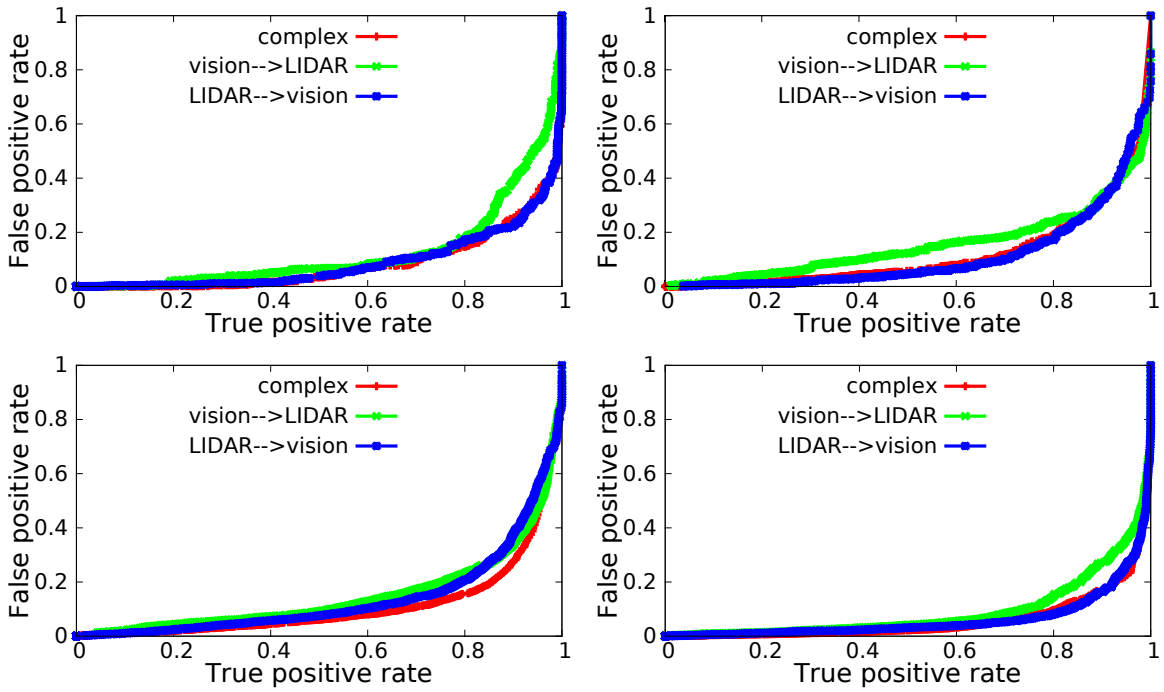


Figure 6.5 – ROCs of fused correspondence detection using the proposed GT data. Multiple graphs illustrates cross-validation procedure. The red curve represents the cross-verified strategy of Eq. 3.8. The remaining curves (blue and green) represent ROCs for uni-modal association mechanism.

(see Fig. 3.9). Moreover, both KITTI and the proposed dataset's ROCs are more smooth than the one of Honda. This is because on KITTI and the proposed dataset, the SOMs learning and tests perform using GT annotations. Raw detections from noisy sources, however, were employed on Honda data.

ROCs for vision–>LIDAR and LIDAR–>vision uni-modal correspondence methods and multi-modal strategies have similar results for datasets using Ground Truth. In that case a good enough uni-modal correspondence can not be significantly improved by multi-modal fusion, because of the same information in two modalities.

In contrast, the SOMs learning using noisy detectors provides modalities with different information. These uni-modal associations are not precise enough because they are learned

with noisy information.

ROCs depend on the number of observed objects at one sampling time. For the case of proposed dataset, the number of observed objects is at maximum 4. For two scenarios is only necessary at maximum 2. That explains better ROCs for the cases where the learning is effectuated using Scenarios 1 and 2.

6.4 Context implementation

The scenarios of the dataset were recorded on a parking lot near the research laboratory, where Open Street maps does not provide detailed cartography. To cope with this, internal vehicle roads were added. Tracked objects agreeing to the contextual information can be identified. Objects disagreeing to contextual prior can be reported as critical. The observed behaviour of the latter set of objects can represent a high risk for autonomous vehicle applications. In the considered use-case, the proposed algorithm analyses four pedestrians and provides "along" and "against" classes of the context motions.

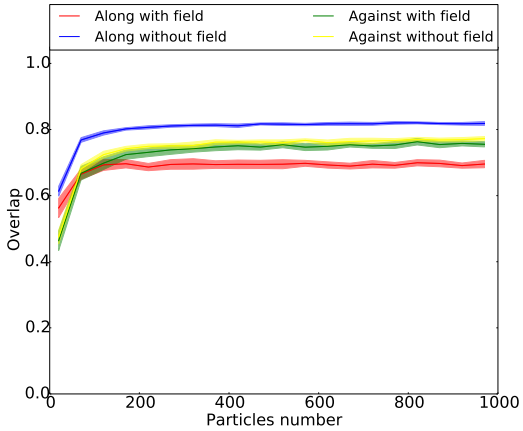
It is possible to evaluate the impact of the contextual information on the tracking process for these two groups of pedestrians. In Fig. 6.6 the performance of the tracking according to overlap, Euclidean distance and continuity criteria is presented.

The tracking is effectuated on ENU space because here the multi-modality or detection nature are not principal. The GPS GT annotations are easier to use for context implementation test because the context is defined also in GPS coordinates. Tracker parameters are the same as for LIDAR-view tracking, described in Sec. 6.6.2. Additive noise parameters applied to GPS observations: $\sigma_c = 0.15$ (m) and $P_{fn} = 0.2$.

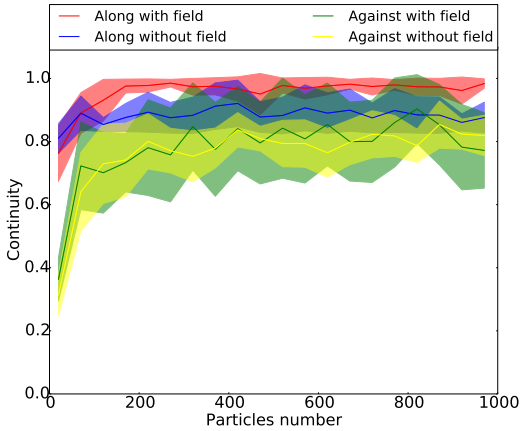
For the same scenario a classification graph for tracks along and against contextual directions is presented at Fig. 6.7. Here additional discretization is applied: the classification value is bigger than 1, i.e. impact of injected particles are bigger, the new classification value is set to 1. If the classification value is smaller than 1, a new classification value is set to -1. This discretization is applied to normalize the criteria.

As a conclusion, our investigations have shown that the context implementation method improves the continuity of tracked objects up to 10%. The overlap and the distance criteria are however impacted since positioning errors increased up to 10 cm (please see the vertical gap between red and blue curves in Fig. 6.6c). The increasing on the track positioning error is induced by the context particles mechanism injection. In detail, the context in the recorded scenario for sidewalks is composed of two possible walking directions (i.e. forward and backward along the sidewalk). Both directions are taken into account in Scenario 4 (see Fig. 6.8). When context particles are injected during filtering, two sets of particles represent the expected context motions. Since a pedestrian can only move in one of the two context directions a set of injected particles misplace the track mean state. This problem can be tackle by restricting the considered context direction to the most probable one. This can be a perspective of this research.

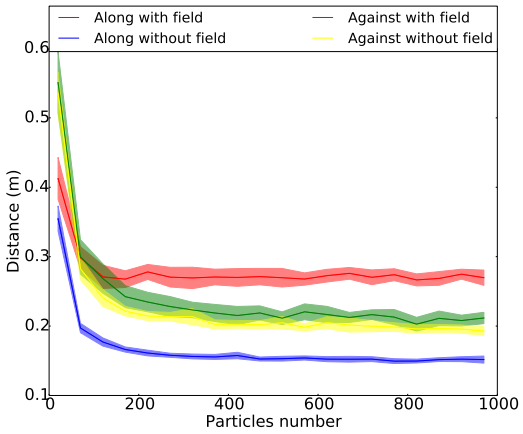
For tracks following the vector field class, the context implementation stabilizes tracking in terms of continuity since the dispersion of continuity values decreases from 0.07 to 0.03 (i.e. 2 times better) in the continuity scale.



(a) Overlap



(b) Continuity



(c) Distance

Figure 6.6 – Comparison of accuracy for tracks moving along and against context vector fields without and with them using a variable model force coefficient. The used data is taken from GPS GT of the scenario 4 of proposed dataset

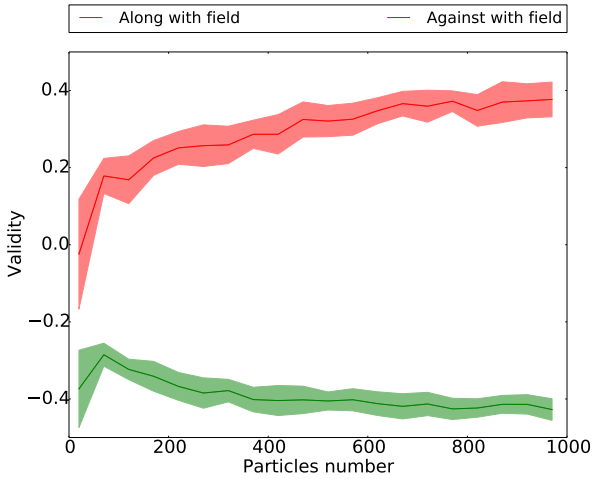


Figure 6.7 – Direction compatibility measurements (validity) for proposed dataset scenario. The values are calculated according to Eq. 4.3 and supplement discretization: for values bigger than 0.0, the movements is assumed to be along the field, and against the field otherwise.

For tracks contradicting the context prior, the continuity remains at the same level. The positioning criterion is slightly impacted of 1-1.5 cm (please see the vertical gap between green and yellow curves in Fig. 6.6c).

Experiments have shown that the quality of direction compatibility classification performs as

expected. Tracks following an unexpected trajectory are detected in a time horizon of 2 seconds (i.e. ≈10 timestamps of GPS sampling - 5Hz). Without lost of generality, let's consider that the LIDAR or camera detections can classify the object's context compatibility at the same rate (i.e. 10 samples). A camera performing object detection at 15 fps would require 660 ms and and a LIDAR at 25 Hz would need 400 ms to detect such a behaviour. A child moving at 24 km/h speed, the classification latency is equivalent to 4.35 m and 2.64 m for camera and LIDAR respectively. In a urban environment, the object tracking system would require a distance of 35 m (25 m braking distance and 660 ms latency) in order to mitigate or avoid such a collision scenario.

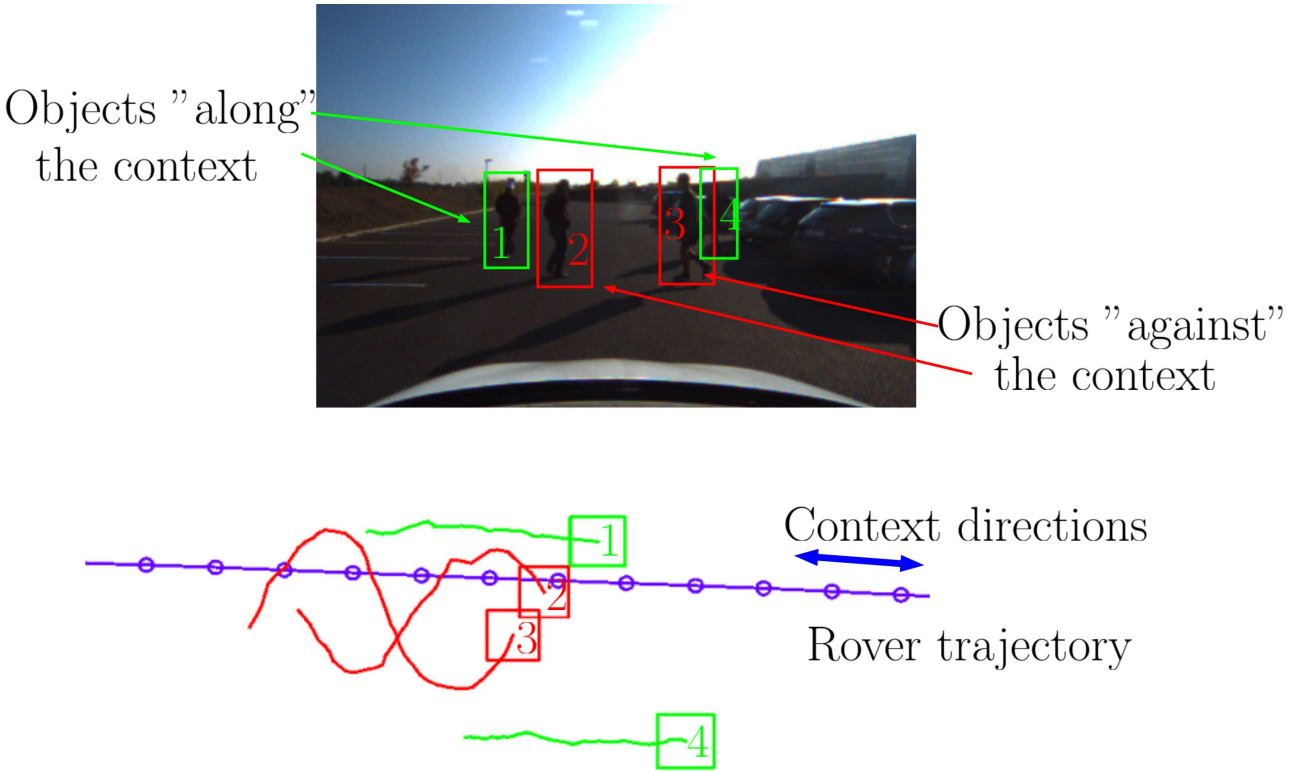


Figure 6.8 – Visualization of the real-time validity evaluation for Scenario 4. Trajectory has a red color when the object is classified as against the context and green color when the object is along the context

Conclusion and perspectives

Conclusion

In thesis a context-aided multi-modal perception system has been studied. The proposed system was intended for tracking moving objects. Our investigations were aimed at determining and quantifying the impact of context information in the performance of a multi-sensor tracking framework. To this end, such a tracking framework was developed with minor contributions. The multi-sensor nature of the considered system lead us also to contribute with a new multi-sensor association strategy. The proposed methods were applied to an intelligent vehicle and they were experimentally validated on full-scale use-cases including semi-urban and urban environments.

The first step of the conducted investigation addressed the well-known problem of tracking mobile surrounding objects from a moving multi-sensor platform. Important efforts have been done in the last decades to address this complex problem, particularly for Intelligent Vehicles (IV) applications. After surveying state-of-the-art techniques, a Monte-Carlo PF and PHD-based multi-object tracking algorithm was implemented taking into account the complexity of the approach, its flexibility to perform multi-sensor data fusion and the adaptability of framework to integrate contextual information at the fusion stage.

Secondly, a new multi-object tracking framework tightly coupled to a contextual information integration mechanism was proposed so as to improve the system performance in terms of accuracy and stability. In detail, prior knowledge was represented as context particles since tracks states are managed by a particle filter. Such particles were introduced at the re-sampling filter stage allowing for tracks behavior classification. That is, the states of tracked objects agreeing with contextual data are stabilized and predictions are improved. The states of tracked objects contradicting contextual data are clearly identified. This last feature can be employed to identify unexpected motions which constitutes a key feature for IV and autonomous vehicles applications.

Later on, multi-sensor data association was investigated. The association of multi-sensor data is often facilitated by the representation (i.e. transformation) of the data into a common reference coordinate system. In consequence, data association methods are subject to errors and uncertainties induced by data transformations. Motivated by the complexity and the sensibility of existing multi-sensor calibration procedures (i.e. extrinsic calibration), a new learning-based method using Self Organizing Maps (SOM) was proposed. This technique is not only transposable to different sensing modalities (e.g. vision, LIDAR, RADAR) but can also model a non-unique association between sensors coordinate systems. This alternative approach

was validated on KITTI dataset and on Honda proprietary data for associating LIDAR - camera information.

Aiming at experimentally validating the previous stated contributions of this thesis on specific use-cases, an important effort was finally devoted for conceiving and building a new multi-modal dataset. The intended contributions of this dataset are to provide :

— Asynchronous measurements of a standard multi-modal IV system including a wide-view monocular vision, a multi-layer LIDAR, wheel speed sensors and a RTK-L1 localization system.

— A Ground Truth reference positioning is available for intelligent vehicle and surrounding objects (i.e. up to 4) enabled by means of a RTK positioning system and post-processed objects trajectories on RTK-L1 mode. These data were also represented on a common reference coordinate system validating the integrity in terms of spatial and temporal coherence of the data. Moreover, geometrical calibration and manually-defined annotations are also included.

— A reference and stand-alone perception means facilitating the development and the performance estimation of IV applications.

To conclude, a thorough analysis of the proposed learning-based multi-sensor data association method was reported as well as for the context-aided multi-modal object tracking system.

Perspectives

The promising results of the learning-based multi-sensor calibration provide new insights regarding alternative strategies allowing for mutual support between different sensing modalities. Such strategies would notably enhance False Alarms (FA) statistics in complex dynamic scenarios. A second but not less important perspective concerns the detection processing stage of the multi-modal perception system being considered as an available input in this study. The diversity of detection methods regarding each perception modality was out of the scope of the proposed investigation. However, tracking-by-detection like strategies would also help to cope with detection errors as proposed by [Gepperth, Ortiz, et al. 2016].

Certainly, some improvements of the proposed multi-sensor data association strategy still remain open. For instance, SOM intrinsic properties may lead to poor quality association because of its node structure spareness particularly for data located near to detection border regions. As a perspective, node-to-node association could be improved into a polygon-to-polygon structure. To this end, the association procedure would detect nearest nodes projecting theirs weights onto a set of nodes into the corresponding space. Such transfered nodes would define a linear space allowing for higher precision and dealing with node-sparse areas. A second limitation is the presence of some ambiguous associations induced by the presence of weights local maxima on SOMs. A possible idea to tackle this limitation is to use not only object position but also tracked objects dynamics in the SOM learning stage.

Concerning the integration of contextual information, an interesting extension of the proposed concept will be its inclusion at an earlier processing stage (e.g. detection) and a multi-hypothesis generation based contextual-priors.

Finally, thanks to the experimental protocol developed for retrieving multi-modal data on dynamic scenes, the proposed dataset can be enhanced by including a larger selection of complex use-cases covering for instance variety of occlusions, intersections, parallel motions and pedestrian groups. The use of inertial sensors and an INS/GNSS positioning system as a reference would definitely facilitate the dynamic scene analysis.

Appendices

Appendix A: ROS messages listings

```
pcan_msgs/CAN
  std_msgs/Header  header
    uint32 seq        //not used
    time   stamp      // POSIX timestamp came from PCs time
                      // at the moment of message creation
    string frame_id  //Encoder's ID
  uint8 lenght        //Length of actual used PCAN message
  uint8[8] data
//PCAN main array with number of impulses, impulse-
                      //calculated speed and intern encoders timestamp
```

Listing A.1 – Encoders message format pcan_msgs/CAN
with main array explained in Sec. 5.4.1

```
std_msgs/Header  header
  uint32  seq       // not used
  time stamp        // POSIX timestamp came from PCs time
                    // at the moment of message creation
  string  frame_id // raw frame of type GPGGA/GNGGA or GPRMC/GNRMC
```

Listing A.2 – GPS message format std_msgs/Header

```
sensor_msgs/Image
  std_msgs/Header  header
    uint32  seq       // not used
    time stamp        // POSIX timestamp came from PCs time
                      // at the moment of message creation
    string frame_id   // not used
  uint32 height
// image height in pixels, that is, number of rows
  uint32 width
// image width in pixels, that is, number of columns
  string encoding     // encoding. Here RGB8 is used
  uint8 is_bigendian //is this data bigendian?
  uint32 step         //Full row length in bytes
  uint8[] data        //actual matrix data, size is (step * rows)
```

Listing A.3 – Vision message format sensor_msgs/Image

```
sensor_msgs/PointCloud2
  std_msgs/Header  header
```

```
    uint32 seq          //not used
    time stamp          // POSIX timestamp came from PCs time
    string frame_id     //not used
  uint32 height         //2D structure of the point cloud. If the cloud
  uint32 width          //is unordered, height is 1 and width is the
                        //length of the point cloud
  sensor_msgs/PointField[] fields //Each point composition
    uint8 INT8=1        //data types
    uint8 UINT8=2
    uint8 INT16=3
    uint8 UINT16=4
    uint8 INT32=5
    uint8 UINT32=6
    uint8 FLOAT32=7
    uint8 FLOAT64=8
    string name         //Name of data type
                        //(x,y,z,timedelta,echowidth,layerechoflags)
    uint32 offset       //Offset from start of point struct
    uint8 datatype      // Datatype enumeration, see above
    uint32 count        //How many elements in the field
  bool is_bigendian     // Is this data bigendian?
  uint32 point_step     //  Length of a point in bytes
  uint32 row_step       // Length of a row in bytes.
                        // It varies from one mesure to another
  uint8[] data          // Actual point data, size is (row_step*height)
  bool is_dense         // True if there are no invalid points
/*The last element in a point description is "layerechoflags",
where 0,1 bits are layer number; 2,3 bits are echo and
4,5,6 bits are flags*/
```

Listing A.4 – LIDAR message format sensor_msgs/PointCloud2

Appendix B: dataset file structure

```
/scenario_name
├── bags ..... Bags folder
│   └── main.bag ..... ROS bag containing all scenario topics and messages
├── gps_raw ..... Raw GSP data captured in offline mode
│   ├── base
│   │   ├── base.SBF ..... APS-3 raw logged data
│   │   └── base.pos ..... NMEA post-processed output
│   ├── rover_reference
│   │   ├── rover_reference.SBF ..... APS-3 raw logged data
│   │   └── rover_reference.pos ..... NMEA post-processed output
│   ├── rover_perception
│   │   ├── rover_perception.ubx ..... UBX raw logged data
│   │   └── rover_perception.pos ..... NMEA post-processed output
│   ├── track0 ..... Raw and NMEA post-processed output for all tracks
│   │   ├── track0.ubx
│   │   └── track0.pos
│   ├── ...
│   └── trackN
│       ├── trackN.ubx
│       └── trackN.pos
├── extracted
│   ├── images ..... PNG images ordered from 0 to M
│   │   ├── cam0000000.png
│   │   ├── ...
│   │   └── camM.png
│   ├── clouds ..... Text scans ordered from 0 to K
│   │   ├── lid0000000.png
│   │   ├── ...
│   │   └── lidK.png
│   ├── cam.txt ..... Timestamps associated with images
│   ├── lid.txt ..... Timestamps associated with scand
│   ├── encoders.txt ..... Encoders messages
│   ├── perception_rover.txt ... Post-processed NMEA with POSIX timestamps
│   ├── reference_rover.txt .... Post-processed NMEA with POSIX timestamps
│   ├── track0 ..... Post-processed NMEA with POSIX timestamps
│   ├── ...
│   └── trackN
├── transforms
│   ├── lidToCam.txt . Rigid transformation from LIDAR space to camera one
│   ├── gpsToLid.txt .... Set of transformations from Pseudo-LIDAR space to
│   │   LIDAR one
│   ├── camera_calibration_raw ..... Raw camera calibration data
│   └── LIDAR_camera_calibration_raw ..... Raw LIDAR-camera calibration data
└── annotations
    └── tracklets.xml . Annotated Ground Truth for tracks positions in GPS,
        LIDAR, camera spaces
```

Figure B.1 – File structure corresponding to one scenario from the dataset

Appendix C: dataset tracklets example

```
<?xml version="1.0" encoding="UTF-8"?>
<Tracklets Type="Multimodal">
    <count>2</count>
    <Track>
        <Video>
            <frame x="219.343" y="344.778" w="33.7319" h="85.8011"
            timestamp="1466601582.944839716"/>
            <frame x="223.818" y="345.049" w="33.8125" h="85.7738"
            timestamp="1466601583.008192301"/>
        </Video>
        <Cloud>
            <frame timestamp="1466601549.930003643">
                <point x="4.81342" y="5.73641" z="-0.156859" layer="0"/>
                <point x="4.9542" y="5.80063" z="-0.159791" layer="1"/>
            </frame>
            <frame timestamp="1466601549.970012188">
                <point x="4.81985" y="5.74407" z="-0.157068"/>
                <point x="4.90226" y="5.73981" z="-0.158115"/>
                <point x="4.81229" y="5.63447" z="-0.0517311"/>
            </frame>
        </Cloud>
        <GPS>
            <frame lat="48.7129" lon="2.16807" alt="205.19" e="-0.796161"
            n="-2.70606" u="0.159001" xrel="0.915924" yrel="-2.6679"
            zrel="0.159001" timestamp="1466601518.079777002"/>
            <frame lat="48.7129" lon="2.16807" alt="205.196" e="-0.799964"
            n="-2.70124" u="0.160001" xrel="0.910034" yrel="-2.66617"
            zrel="0.160001" timestamp="1466601518.279969931"/>
        </GPS>
    </Track>
    <Track>
        <Video>
             <frame x="977.951" y="365.64" w="80.417" h="140.541"
             timestamp="1466601687.010539055"/>
        </Video>
        <Cloud>
            <frame timestamp="1466601548.451144934">
                <point x="8.28533" y="9.87407" z="-0.0899884" layer="2"/>
            </frame>
        </Cloud>
        <GPS>
            <frame lat="48.713" lon="2.16809" alt="205.116" e="0.405501"
            n="1.29742" u="0.0850001" xrel="-0.419744" yrel="1.29289"
            zrel="0.0850001" timestamp="1466601518.079777002"/>
        </GPS>
    </Track>
</Tracklets>
```

Listing C.1 – Example of an XML-format for tracks annotations

Publications

Gepperth, Alexander, Michael Garcia Ortiz, Egor Sattarov, and Bernd Heisele (2016). "Dynamic attention priors: a new and efficient concept for improving object detection". In: *Neurocomputing* 197, pp. 14–28. ISSN: 0925-2312. DOI: http://dx.doi.org/10.1016/j.neucom.2016.01.036. URL: http://www.sciencedirect.com/science/article/pii/S0925231216001259.

Gepperth, Alexander, Egor Sattarov, Bernd Heisele, and Sergio Alberto Rodriguez Flores (2014). "Robust visual pedestrian detection by tight coupling to tracking". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1935–1940. DOI: 10.1109/ITSC.2014.6957989.

Sattarov, Egor, Sergio Alberto Rodríguez Florez, Alexander Gepperth, and Roger Reynaud (2014). "Context-based vector fields for multi-object tracking in application to road traffic". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1179–1185. DOI: 10.1109/ITSC.2014.6957847.

Sattarov, Egor, Alexander Gepperth, Sergio Alberto Rodriguez Florez, and Roger Reynaud (2015). "Calibration-free match finding between vision and LIDAR". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1061–1067. DOI: 10.1109/IVS.2015.7225825.

References

Abdulhafiz, Waleed A. and Alaa Khamis (2013). "Bayesian approach to multisensor data fusion with Pre- and Post-Filtering". In: *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*, pp. 373–378. DOI: 10.1109/ICNSC.2013.6548766 (cit. on p. 37).

Alameda-Pineda, Xavier, Jacopo Staiano, Ramanathan Subramanian, Ligia Maria Batrinca, Elisa Ricci, Bruno Lepri, Oswald Lanz, and Nicu Sebe (2015). "SALSA: A Novel Dataset for Multimodal Group Behavior Analysis". In: *CoRR* abs/1506.06882. URL: http://arxiv.org/abs/1506.06882 (cit. on p. 85).

Amditis, Angelos, George Thomaidis, Pantelis Maroudis, Panagiotis Lytrivis, and Giannis Karaseitanidis (2012). "Multiple Hypothesis Tracking Implementation". In: *Laser Scanner Technology*. InTech. Chap. 10. DOI: 10.5772/33583. URL: http://www.intechopen.com/books/laser-scanner-technology/multiple-hypothesis-tracking-implementation (cit. on p. 45).

APS-3 User manual (2011). 2.0. ALTUS Positioning Systems Inc. URL: http://www.surveysoft.it/SOFTWARE/APS-3%20User%20Manual%20Rev%202.00.pdf (cit. on p. 92).

Arulampalam, M. Sanjeev, Simon Maskell, Neil Gordon, and Tim Clapp (2002). "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking". In: *IEEE Transactions on Signal Processing* 50.2, pp. 174–188. ISSN: 1053-587X. DOI: 10.1109/78.978374 (cit. on p. 47).

Avidan, Shai (2001). "Support Vector Tracking". In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1, pages. DOI: 10.1109/CVPR.2001.990474 (cit. on p. 43).

Baltieri, Davide, Roberto Vezzani, and Rita Cucchiara (2011). "3DPeS: 3D People Dataset for Surveillance and Forensics". In: *Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding*. J-HGBU '11. Scottsdale, Arizona, USA: ACM, pp. 59–64. ISBN: 978-1-4503-0998-1. DOI: 10.1145/2072572.2072590. URL: http://doi.acm.org/10.1145/2072572.2072590 (cit. on p. 85).

Bar-Shalom, Yaakov and Edison Tse (1975). "Tracking in a cluttered environment with probabilistic data association". In: *Automatica* 11.5, pp. 451–460. ISSN: 0005-1098. DOI: http://dx.doi.org/10.1016/0005-1098(75)90021-7. URL: http://www.sciencedirect.com/science/article/pii/0005109875900217 (cit. on p. 46).

Bazzani, Loris, Domenico Bloisi, and Vittorio Murino (2009). "A Comparison of Multi Hypothesis Kalman Filter and Particle Filter for Multi-target Tracking". In: *Performance Evaluation of Tracking and Surveillance workshop at CVPR*. Miami, Florida, pp. 47–54 (cit. on p. 46).

Berclaz, Jérôme, François Fleuret, Engin Türetken, and Pascal Fua (2011). "Multiple Object Tracking Using K-Shortest Paths Optimization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.9, pp. 1806–1819. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2011.21 (cit. on p. 85).

Bertalmio, Marcelo, Guillermo Sapiro, and Gregory Randall (1998). "Morphing active contours: a geometric approach to topology-independent image segmentation and tracking". In: *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, 318–322 vol.3. DOI: 10.1109/ICIP.1998.999021 (cit. on p. 43).

Bertozzi, Massimo, Alberto Broggi, Alessandra Fascioli, Thorsten Graf, and Marc-Michael Meinecke (2004). "Pedestrian detection for driver assistance using multiresolution infrared vision". In: *IEEE Transactions on Vehicular Technology* 53.6, pp. 1666–1678. ISSN: 0018-9545. DOI: 10.1109/TVT.2004.834878 (cit. on p. 25).

Bevilacqua, Alessandro, Luigi Di Stefano, and Stefano Vaccari (2005). "Occlusion Robust Vehicle Tracking based on SOM (Self-Organizing Map)". In: *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*. Vol. 2, pp. 84–89. DOI: 10.1109/ACVMOT.2005.87 (cit. on p. 62).

Black, Michael J. and Allan D. Jepson (1998). "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation". In: *International Journal of Computer Vision* 26.1, pp. 63–84. ISSN: 1573-1405. DOI: 10.1023/A:1007939232436. URL: http://dx.doi.org/10.1023/A:1007939232436 (cit. on p. 43).

Blackman, Samuel S. (2004). "Multiple hypothesis tracking for multiple target tracking". In: *IEEE Aerospace and Electronic Systems Magazine* 19.1, pp. 5–18. ISSN: 0885-8985. DOI: 10.1109/MAES.2004.1263228 (cit. on pp. 45, 74).

Bohlooli, Ali and Kamal Jamshidi (2012). "A GPS-free method for vehicle future movement directions prediction using SOM for VANET". In: *Applied Intelligence* 36.3, pp. 685–697. ISSN: 1573-7497. DOI: 10.1007/s10489-011-0289-9. URL: http://dx.doi.org/10.1007/s10489-011-0289-9 (cit. on p. 61).

Bonarini, Andrea, Wolfram Burgard, Giulio Fontana, Matteo Matteucci, Domenico Giorgio Sorrenti, and Juan Domingo Tardos (2006). "RAWSEEDS: Robotics Advancement through Web-publishing of Sensorial and Elaborated Extensive Data Sets". In: *In proceedings of IROS'06 Workshop on Benchmarks in Robotics Research*. URL: http://www.robot.uji.es/EURON/en/iros06.htm (cit. on p. 86).

Bouaziz, Samir (2013). *Véhicule automatique - Étude technique*. Tech. rep. CAR&D, IEF (cit. on p. 89).

Bouguet, Jean-Yves (2003). *Camera Calibration Toolbox for Matlab*. URL: http://robots.stanford.edu/cs223b04/JeanYvesCalib/index.html (cit. on p. 96).

Bradski, Gary (2000). "OpenCV tools". In: *Dr. Dobb's Journal of Software Tools* (cit. on p. 96).

Brščić, Dražen, Takayuki Kanda, Tetsushi Ikeda, and Takahiro Miyashita (2013). "Person position and body direction tracking in large public spaces using 3D range sensors". In: *IEEE Transactions on Human-Machine Systems* (cit. on p. 85).

Burri, Michael, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W. Achtelik, and Roland Siegwart (2016). "The EuRoC micro aerial

vehicle datasets". In: *The International Journal of Robotics Research*. DOI: 10.1177/0278364915620033. eprint: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.full.pdf+html. URL: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract (cit. on p. 86).

Cai, Ziyun, Jungong Han, Li Liu, and Ling Shao (2016). "RGB-D datasets using microsoft kinect or similar sensors: a survey". In: *Multimedia Tools and Applications*, pp. 1–43. ISSN: 1573-7721. DOI: 10.1007/s11042-016-3374-6. URL: http://dx.doi.org/10.1007/s11042-016-3374-6 (cit. on p. 85).

Caraffi, Claudio, Tomas Vojir, Jura Trefny, Jan Sochman, and Jiri Matas (2012). "A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera". In: *ITS Conference*, pp. 975–982 (cit. on p. 85).

Caron, Louis-Charles, Yang Song, David Filliat, and Alexander Gepperth (2014). "Neural network based 2D/3D fusion for robotic object recognition". In: *Proc. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)* (cit. on p. 34).

Castanedo, Federico (2013). "A Review of Data Fusion Techniques". en. In: *The Scientific World Journal* 2013. DOI: 10.1155/2013/704504. URL: http://www.hindawi.com/journals/tswj/2013/704504/abs/ (cit. on p. 32).

Ceriani, Simone, Giulio Fontana, Alessandro Giusti, Daniele Marzorati, Matteo Matteucci, Davide Migliore, Davide Rizzi, Domenico G. Sorrenti, and Pierluigi Taddei (2009). "Rawseeds ground truth collection systems for indoor self-localization and mapping". In: *Autonomous Robots* 27.4, pp. 353–371. DOI: 10.1007/s10514-009-9156-5. URL: http://www.springerlink.com/content/k924032g72818h53/ (cit. on p. 86).

CGGBP.35.6.A.02 Specification datasheet (2015). Taoglas. URL: https://taoglas.com/images/product_images/original_images/CGGBP.35.6.A.02.pdf (cit. on p. 93).

Chang, Ting-Hsun and Shaogang Gong (2001). "Tracking multiple people with a multi-camera system". In: *Multi-Object Tracking, 2001. Proceedings. 2001 IEEE Workshop on*, pp. 19–26. DOI: 10.1109/MOT.2001.937977 (cit. on p. 25).

Chapuis, Roland, Romuald Aufrere, and Frédéric Chausse (2002). "Accurate road following and reconstruction by computer vision". In: *IEEE Transactions on Intelligent Transportation Systems* 3.4, pp. 261–270. ISSN: 1524-9050. DOI: 10.1109/TITS.2002.804751 (cit. on p. 75).

Chavez-Garcia, R. Omar (2014). "Multiple Sensor Fusion for Detection, Classification and Tracking of Moving Objects in Driving Environments". Theses. Université de Grenoble. URL: https://hal.archives-ouvertes.fr/tel-01082021 (cit. on p. 30).

Cheikh, Faouzi Alaya, Sajib Kumar Saha, Victoria Rudakova, and Peng Wang (2012). "Multi-people tracking across multiple cameras". In: *International Journal of New Computer Architectures and their Applications (IJNCAA)*, 23–33 vol.2 (cit. on p. 40).

Cheng, Yang and Tarunraj Singh (2007). "Efficient particle filtering for road-constrained target tracking". In: *IEEE Transactions on Aerospace and Electronic Systems* 43.4, pp. 1454–1469. ISSN: 0018-9251. DOI: 10.1109/TAES.2007.4441751 (cit. on p. 75).

Cho, Hyungg, Young-Woo Seo, B. V. K. Vijaya Kumar, and Ragunathan Raj Rajkumar (2014). "A multi-sensor fusion system for moving object detection and tracking in urban driving environments". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1836–1843. DOI: 10.1109/ICRA.2014.6907100 (cit. on pp. 17, 24, 34, 38, 85).

Cho, Jeongho, Jose C. Principe, Deniz Erdogmus, and Mark A. Motter (2006). "Modeling and inverse controller design for an unmanned aerial vehicle based on the self-organizing map". In: *IEEE Transactions on Neural Networks* 17.2, pp. 445–460. ISSN: 1045-9227. DOI: 10.1109/TNN.2005.863422 (cit. on p. 61).

Chumerin, Nikolay and Marc M. Van Hulle (2008). "Cue and Sensor Fusion for Independent Moving Objects Detection and Description in Driving Scenes". In: *Signal Processing Techniques for Knowledge Extraction and Information Fusion*. Boston, MA: Springer US, pp. 161–180. ISBN: 978-0-387-74367-7. DOI: 10.1007/978-0-387-74367-7_9. URL: http://dx.doi.org/10.1007/978-0-387-74367-7_9 (cit. on p. 30).

Comaniciu, Dorin and Peter Meer (2002). "Mean shift: a robust approach toward feature space analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5, pp. 603–619. ISSN: 0162-8828. DOI: 10.1109/34.1000236 (cit. on p. 42).

Cox, David and Deborah G. Mayo (2010). "Objectivity and Conditionality in Frequentist Inference". In: *Error and Inference: Recent Exchanges on Experimental Reasoning, Reliability, and the Objectivity and Rationality of Science*. Ed. by Deborah G. Mayo and Aris Spanos. Cambridge University Press, p. 276 (cit. on p. 35).

Cox, Ingemar J. (1992). "A review of statistical data association techniques for motion correspondence". In: *International Journal of Computer Vision* 10 (cit. on p. 46).

Cramer, Heiko, Ullrich Scheunert, and Gerd Wanielik (2003). "Multi sensor fusion for object detection using generalized feature models". In: *Information Fusion, 2003. Proceedings of the Sixth International Conference of*. Vol. 1, pp. 2–10. DOI: 10.1109/ICIF.2003.177419 (cit. on p. 35).

Crowley, James L. (1993). "Principles and Techniques for Sensor Data Fusion". In: *Multisensor Fusion for Computer Vision*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 15–36. ISBN: 978-3-662-02957-2. DOI: 10.1007/978-3-662-02957-2_2. URL: http://dx.doi.org/10.1007/978-3-662-02957-2_2 (cit. on pp. 23, 32).

Dalal, Navneet and Bill Triggs (2005). "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177 (cit. on p. 35).

Darms, Michael, Paul Rybski, and Christopher Urmson (2008). "A Multisensor Multiobject Tracking System for an Autonomous Vehicle Driving in an Urban Environment". In: *AVEC 2008 - 9th International Symposium on Advanced Vehicle Control*. Kobe (cit. on pp. 17, 26).

Davis, James W. and Vinay Sharma (2007). "Background-subtraction using contour-based fusion of thermal and visible imagery". In: *Computer Vision and Image Understanding* 106.2–3. Special issue on Advances in Vision Algorithms and Systems beyond the Visible Spectrum, pp. 162–182. ISSN: 1077-3142. DOI: http://dx.doi.org/10.1016/j.cviu.2006.06.010. URL: http://www.sciencedirect.com/science/article/pii/S1077314206001834 (cit. on p. 34).

Denoeux, Thierry (2006). "The cautious rule of combination for belief functions and some extensions". In: *2006 9th International Conference on Information Fusion*, pp. 1–8. DOI: 10.1109/ICIF.2006.301572 (cit. on p. 38).

Derder, Abdessamed and Samira Moussaoui (2014). "Target Tracking in VANETs Using V2I and V2V Communication". In: *Advanced Networking Distributed Systems and Applications (INDS), 2014 International Conference on*, pp. 19–24. DOI: 10.1109/INDS.2014.11 (cit. on p. 32).

Derder, Abdessamed, Samira Moussaoui, and Abdelwahab Boualouache (2015). "Target tracking in VANETs using V2V communication with packet load enhancement". In: *New Technologies of Information and Communication (NTIC), 2015 First International Conference on*, pp. 1–6. DOI: 10.1109/NTIC.2015.7368738 (cit. on p. 32).

Dickmanns, Ernst Dieter (2003). "An Advanced Vision System for Ground Vehicles". In: *In International Workshop on In-Vehicle Cognitive Computer Vision Systems (IVC2VS)* (cit. on p. 25).

Dixon, Kevin (2006). "StarFire: A Global SBAS for Sub-Decimeter Precise Point Positioning". In: *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*. Fort Worth, TX, pp. 2286–2296 (cit. on p. 30).

Dollár, Piotr, Christian Wojek, Bernt Schiele, and Pietro Perona (2009). "Pedestrian Detection: A Benchmark". In: *CVPR* (cit. on p. 85).

— (2012). "Pedestrian Detection: An Evaluation of the State of the Art". In: *PAMI* 34 (cit. on p. 85).

Dubuisson, Séverine and Christophe Gonzales (2016). "A survey of datasets for visual tracking". In: *Machine Vision and Applications* 27.1, pp. 23–52. ISSN: 1432-1769. DOI: 10.1007/s00138-015-0713-y. URL: http://dx.doi.org/10.1007/s00138-015-0713-y (cit. on p. 85).

Esteban, Jaime, Andrew Starr, Robert Willetts, Paul Hannah, and Peter Bryanston-Cross (2005). "A Review of data fusion models and architectures: towards engineering guidelines". In: *Neural Computing & Applications* 14.4, pp. 273–281. ISSN: 1433-3058. DOI: 10.1007/s00521-004-0463-7. URL: http://dx.doi.org/10.1007/s00521-004-0463-7 (cit. on pp. 23, 33).

Fienberg, Stephen E. (2006). "When did Bayesian inference become "Bayesian"?" In: *BAYESIAN ANALYSIS*, pp. 1–41 (cit. on p. 35).

Flea 2 Technical Reference Manual (2011). 1.10. Point Grey Research. URL: https://www.ptgrey.com/support/downloads/10117 (cit. on p. 90).

Fleuret, François, Jérôme Berclaz, Richard Lengagne, and Pascal Fua (2008). "Multi-Camera People Tracking with a Probabilistic Occupancy Map". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2, pp. 267–282 (cit. on p. 85).

Fremont, Vincent, Sergio Alberto Rodriguez Florez, and Philippe Bonnifait (2012). "Circular Targets for 3D Alignment of Video and Lidar Sensors". In: *Advanced Robotics, Taylor & Francis* 26 (18), pp. 2087–2113 (cit. on pp. 39, 96).

Fritsch, Jannik, Tobias Kuehnl, and Andreas Geiger (2013). "A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms". In: *IEEE 16th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1693–1700 (cit. on p. 53).

Geiger, Andreas, Philip Lenz, Christoph Stiller, and Raquel Urtasun (2013). "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* 32, pp. 1231–1237 (cit. on pp. 31, 86).

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074 (cit. on pp. 67, 86).

Geiger, Andreas, Frank Moosmann, Oemer Car, and Bernhard Schuster (2012). "Automatic Calibration of Range and Camera Sensors using a single Shot". In: *International Conference on Robotics and Automation (ICRA)* (cit. on p. 39).

Gepperth, Alexander, Michael Garcia Ortiz, Egor Sattarov, and Bernd Heisele (2016). "Dynamic attention priors: a new and efficient concept for improving object detection". In: *Neurocomputing* 197, pp. 14–28. ISSN: 0925-2312. DOI: http://dx.doi.org/10.1016/j.neucom.2016.01.036. URL: http://www.sciencedirect.com/science/article/pii/S0925231216001259 (cit. on pp. 53, 123).

Gepperth, Alexander, Egor Sattarov, Bernd Heisele, and Sergio Alberto Rodriguez Flores (2014). "Robust visual pedestrian detection by tight coupling to tracking". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1935–1940. DOI: 10.1109/ITSC.2014.6957989 (cit. on p. 53).

Ghahroudi, Mahdi Rezaei and Alireza Fasih (2007). "A Hybrid Method in Driver and Multisensor Data Fusion, Using a Fuzzy Logic Supervisor for Vehicle Intelligence". In: *Proceedings of the 2007 International Conference on Sensor Technologies and Applications*. SENSORCOMM '07. Washington, DC, USA: IEEE Computer Society, pp. 393–398. ISBN: 0-7695-2988-7. DOI: 10.1109/SENSORCOMM.2007.9. URL: http://dx.doi.org/10.1109/SENSORCOMM.2007.9 (cit. on p. 35).

Goubet, Emmanuel, Joseph Katz, and Fatih Porikli (2006). "Pedestrian tracking using thermal infrared imaging". In: *Proceedings of ELM-2014 Volume 2*. Vol. 6206, pages. DOI: 10.1117/12.673132. URL: http://dx.doi.org/10.1117/12.673132 (cit. on p. 25).

Grandbois, Brett and Fred Pauling (2012). *Driver node for Sick LD-MRS*. URL: http://wiki.ros.org/sick_ldmrs (cit. on p. 104).

Grewal, Mohinder S. (2011). "Kalman Filtering". In: *International Encyclopedia of Statistical Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 705–708. ISBN: 978-3-642-04898-2. DOI: 10.1007/978-3-642-04898-2_321. URL: http://dx.doi.org/10.1007/978-3-642-04898-2_321 (cit. on p. 44).

Gustafsson, Fredrik (2005). "Statistical signal processing for automotive safety systems". In: *IEEE/SP 13th Workshop on Statistical Signal Processing, 2005*, pp. 1428–1435. DOI: 10.1109/SSP.2005.1628820 (cit. on p. 33).

Habtemariam, Biruk K., Ratnasingham Tharmarasa, Thayananthan Thayaparan, Mahendra Mallick, and Thiagalingam Kirubarajan (2013). "A Multiple-Detection Joint Probabilistic

Data Association Filter". In: *IEEE Journal of Selected Topics in Signal Processing* 7.3, pp. 461–471. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2013.2256772 (cit. on p. 74).

Halfacree, Gareth and Eben Upton (2012). *Raspberry Pi User Guide*. 1st. Wiley Publishing. ISBN: 111846446X, 9781118464465 (cit. on p. 89).

Hall, David Lee, Martin E. Liggins, and James Llinas (2009). *Handbook of multisensor data fusion : theory and practice*. English. 2nd ed. Previous ed.: 2001. Boca Raton, FL : CRC Press. ISBN: 9781420053081 (hbk. : alk. paper). URL: http://trove.nla.gov.au/work/27487296 (cit. on pp. 23, 33, 60).

Hassan, Ehtesham, Gautam Shroff, and Puneet Agarwal (2015). "Multi-sensor Event Detection Using Shape Histograms". In: *Proceedings of the Second ACM IKDD Conference on Data Sciences*. CoDS '15. Bangalore, India: ACM, pp. 20–29. ISBN: 978-1-4503-3436-5. DOI: 10.1145/2732587.2732591. URL: http://doi.acm.org/10.1145/2732587.2732591 (cit. on p. 34).

Held, David, Jesse Levinson, and Sebastian Thrun (2013). "Precision tracking with sparse 3D and dense color 2D data". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 1138–1145. DOI: 10.1109/ICRA.2013.6630715 (cit. on p. 24).

Hendzel, Zenon (2005). "Collision free path planning and control of wheeled mobile robot using Kohonen self-organising map". In: *Technical Sciences* 53.1 (cit. on p. 61).

Heng, Lionel, Bo Li, and Marc Pollefeys (2013). "CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1793–1800. DOI: 10.1109/IROS.2013.6696592 (cit. on p. 25).

Hermes, Christoph, Christian Wöhler, Konrad Schenk, and Franz Kummert (2009). "Long-term Vehicle Motion Prediction". In: *IEEE Intelligent Vehicles Symposium*. Xi'an, China, pp. 652–657 (cit. on p. 47).

Heskes, Tom (1999). *Energy Functions for Self-Organizing Maps*. Ed. by E Oja and S Kaski (cit. on p. 73).

Himmelsbach, Michael, Andre Mueller, Thorsten Luettel, and Hans-Joachim Wuensche (2008). "LIDAR-based 3D Object Perception". In: *Proceedings of 1st International Workshop on Cognition for Technical Systems*. Munich. URL: http://www.velodynelidar.com/lidar/hdlpressroom/pdf/papers/journal_papers/LIDAR-based%203D%20Object%20Perception.pdf (cit. on p. 26).

Holt, Ryan S., Peter A. Mastromarino, Edward K. Kao, and Michael B. Hurley (2010). "Information theoretic approach for performance evaluation of multi-class assignment systems". In: *Proc. SPIE* 7697, pages. DOI: 10.1117/12.851019. URL: http://dx.doi.org/10.1117/12.851019 (cit. on p. 54).

Hsu, Stephen, Sunil Acharya, Abbas Rafii, and Richard New (2006). "Performance of a Time-of-Flight Range Camera for Intelligent Vehicle Safety Applications". In: *Advanced Microsystems for Automotive Applications 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 205–219. ISBN: 978-3-540-33410-1. DOI: 10.1007/3-540-33410-6_16. URL: http://dx.doi.org/10.1007/3-540-33410-6_16 (cit. on p. 27).

Hu, Tao, Stefano Messelodi, and Oswald Lanz (2015). "Dynamic Task Decomposition for Decentralized Object Tracking in Complex Scenes". In: *Comput. Vis. Image Underst.* 134.C, pp. 89–104. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2015.02.007. URL: http://dx.doi.org/10.1016/j.cviu.2015.02.007 (cit. on p. 85).

Hu, Tao, Sinan Mutlu, and Oswald Lanz (2013). "Multicamera People Tracking Using a Locus-based Probabilistic Occupancy Map". In: *Image Analysis and Processing – ICIAP 2013: 17th International Conference, Naples, Italy, September 9-13, 2013, Proceedings, Part II*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 693–702. ISBN: 978-3-642-41184-7. DOI: 10.1007/978-3-642-41184-7_70. URL: http://dx.doi.org/10.1007/978-3-642-41184-7_70 (cit. on p. 85).

Hu, Xiao, Sergio Alberto Rodríguez Florez, and Alexander Gepperth (2014). "A multi-modal system for road detection and segmentation". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 1365–1370. DOI: 10.1109/IVS.2014.6856466 (cit. on p. 75).

Hu, Zhentao, Yong Jin, Jie Li, and Xianxing Liu (2012). "Maneuvering Target Tracking Algorithm Based on Multiple Model Rao-Blackwellised Particle Filter". In: *Journal of Information and Computational Science* 8 (cit. on p. 75).

Humphreys, James and Andrew Hunter (2009). "Multiple object tracking using a neural cost function". In: *Image and Vision Computing* 27.4, pp. 417–424. ISSN: 0262-8856. DOI: http://dx.doi.org/10.1016/j.imavis.2008.06.002. URL: http://www.sciencedirect.com/science/article/pii/S0262885608001352 (cit. on p. 62).

Inductive sensors (2013). Ax-62x-04. Contrinex DW. URL: https://www.rapidonline.com/pdf/50-3739_v1.pdf (cit. on p. 89).

International GNSS Service (IGS), RINEX Working Group and Radio Technical Commission for Maritime Services Special Committee 104 (RTCM-SC104) (2013). *RINEX The Receiver Independent Exchange Format*. version 3.02. URL: ftp://igs.org/pub/data/format/rinex302.pdf (cit. on p. 94).

Isard, Michael and Andrew Blake (1998). "CONDENSATION—Conditional Density Propagation for Visual Tracking". In: *International Journal of Computer Vision* 29.1, pp. 5–28. ISSN: 1573-1405. DOI: 10.1023/A:1008078328650. URL: http://dx.doi.org/10.1023/A:1008078328650 (cit. on p. 43).

Jaechan, Lim (2006). *The Joint Probabilistic Data Association Filter (JPDAF) for Multi-Target Tracking*. Tech. rep. Stony Brook University (cit. on p. 74).

Jafari, Omid Hosseini, Dennis Mitzel, and Bastian Leibe (2014). "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5636–5643. DOI: 10.1109/ICRA.2014.6907688 (cit. on p. 27).

Jain, Anil K. and Richard C. Dubes (1988). *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. ISBN: 0-13-022278-X (cit. on p. 62).

Javed, Omar, Zeeshan Rasheed, Khurram Shafique, and Mubarak Shah (2003). "Tracking across multiple cameras with disjoint views". In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 952–957 vol.2. DOI: 10.1109/ICCV.2003.1238451 (cit. on p. 39).

Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas (2012). "Tracking-Learning-Detection". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 34.7, pp. 1409–1422. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2011.239. URL: http://dx.doi.org/10.1109/TPAMI.2011.239 (cit. on p. 85).

Kang, Jinman, Isaac Cohen, and Gerard Medioni (2004). "Object Reacquisition Using Invariant Appearance Model". In: *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4 - Volume 04*. ICPR '04. Washington, DC, USA: IEEE Computer Society, pp. 759–762. ISBN: 0-7695-2128-2. DOI: 10.1109/ICPR.2004.633. URL: http://dx.doi.org/10.1109/ICPR.2004.633 (cit. on p. 43).

Kaplan, Elliott (2005). *Understanding GPS - Principles and applications*. 2nd edition. Artech House (cit. on p. 28).

Katsarakis, Nikos, Fotios Talantzis, Aristodemos Pnevmatikakis, and Lazaros Polymenakos (2008). "Multimodal Technologies for Perception of Humans". In: *International Evaluation Workshops CLEAR 2007 and RT 2007, Baltimore, MD, USA, May 8-11, 2007, Revised Selected Papers*. Ed. by Rainer Stiefelhagen, Rachel Bowers, and Jonathan Fiscus. Berlin, Heidelberg: Springer-Verlag. Chap. The AIT 3D Audio / Visual Person Tracker for CLEAR 2007, pp. 35–46. ISBN: 978-3-540-68584-5. DOI: 10.1007/978-3-540-68585-2_2. URL: http://dx.doi.org/10.1007/978-3-540-68585-2_2 (cit. on p. 86).

Keller, Christoph Gustav, Markus Enzweiler, and Dariu M. Gavrila (2011). "A New Benchmark for Stereo-based Pedestrian Detection". In: *IEEE Intelligent Vehicles Symposium (IV)* (cit. on p. 86).

Khairdoost, Nima, S. Amirhassan Monadjemi, and Kamal Jamshidi (2013). "Front and Rear Vehicle Detection Using Hypothesis Generation and Verification". In: *Signal & Image Processing*, pp. 31–50. ISSN: 2229-3922. DOI: 10.5121/sipij.2013.4403 (cit. on p. 35).

Kim, Chanho, Fuxin Li, Arridhana Ciptadi, and James M. Rehg (2015). "Multiple Hypothesis Tracking Revisited". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV '15. Washington, DC, USA: IEEE Computer Society, pp. 4696–4704. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.533. URL: http://dx.doi.org/10.1109/ICCV.2015.533 (cit. on p. 45).

Kim, Kyungnam and Larry S. Davis (2006). "Multi-camera Tracking and Segmentation of Occluded People on Ground Plane Using Search-Guided Particle Filtering". In: *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part III*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 98–109. ISBN: 978-3-540-33837-6. DOI: 10.1007/11744078_8. URL: http://dx.doi.org/10.1007/11744078_8 (cit. on p. 39).

Klausne, Andreas, Allan Tengg, and Bernhard Rinner (2007). "Vehicle Classification on Multi-Sensor Smart Cameras Using Feature- and Decision-Fusion". In: *2007 First ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 67–74. DOI: 10.1109/ICDSC.2007.4357507 (cit. on p. 38).

Klein, Dominik A. (2010). *BoBot - Bonn benchmark on tracking*. URL: http://www.iai.uni-bonn.de/~kleind/tracking/index.htm (cit. on p. 85).

Kohonen, Teuvo (1982). "Self-organized formation of topologically correct feature maps". In: *Biological Cybernetics* 43.1, pp. 59–69. ISSN: 1432-0770. DOI: 10.1007/BF00337288. URL: http://dx.doi.org/10.1007/BF00337288 (cit. on p. 61).

Kolb, Andreas, Erhardt Barth, Reinhard Koch, and Rasmus Larsen (2010). "Time-of-Flight Cameras in Computer Graphics". In: *Computer Graphics Forum* 29.1, pp. 141–159. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2009.01583.x. URL: http://dx.doi.org/10.1111/j.1467-8659.2009.01583.x (cit. on p. 27).

Konstantinova, Pavlina, Alexander Udvarev, and Tzvetan Semerdjiev (2003). "A Study of a Target Tracking Algorithm Using Global Nearest Neighbor Approach". In: *Proceedings of the 4th International Conference Conference on Computer Systems and Technologies: E-Learning*. CompSysTech '03. Rousse, Bulgaria: ACM, pp. 290–295. ISBN: 954-9641-33-3. DOI: 10.1145/973620.973668. URL: http://doi.acm.org/10.1145/973620.973668 (cit. on p. 51).

Kooij, Julian F. P., Nicolas Schneider, and Dariu M. Gavrila (2014). "Analysis of pedestrian dynamics from a vehicle perspective". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 1445–1450. DOI: 10.1109/IVS.2014.6856505 (cit. on p. 75).

Koschorrek, Philipp, Tommaso Piccini, Per Öberg, Michael Felsberg, Lars Nielsen, and Rudolf Mester (2013). "A Multi-sensor Traffic Scene Dataset with Omnidirectional Video". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 727–734. DOI: 10.1109/CVPRW.2013.110 (cit. on p. 87).

Kühnlenz, Kolja (2007). *Aspects of Multi-focal Vision*. Fortschrittberichte VDI / 8. VDI-Verlag. ISBN: 9783185129087. URL: https://books.google.fr/books?id=pzaEMwAACAAJ (cit. on p. 25).

Laser Measurement Sensor LD-MRS Operating instructions (2014). SICK AG. URL: https://www.sick.com/media/dox/3/03/803/Operating_instructions_LD_MRS_Laser_Measurement_Sensor_en_IM0032803.PDF (cit. on p. 92).

Leal-Taixé, Laura, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler (2015). "MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking". In: *arXiv:1504.01942 [cs]*. arXiv: 1504.01942. URL: http://arxiv.org/abs/1504.01942 (cit. on p. 85).

Levinson, Jesse and Sebastian Thrun (2013). "Automatic Online Calibration of Cameras and Lasers". In: *Robotics: Science and Systems* (cit. on p. 39).

Li, Longzhen, Tahir Nawaz, and James Ferryman (2015). "PETS 2015: Datasets and challenge". In: *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, pp. 1–6. DOI: 10.1109/AVSS.2015.7301741 (cit. on p. 85).

Li, Xinchao, Martha Larson, and Alan Hanjalic (2015). "Pairwise geometric matching for large-scale object retrieval". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5153–5161. DOI: 10.1109/CVPR.2015.7299151 (cit. on p. 34).

Linder, Timm, Stefan Breuers, Bastian Leibe, and Kai O. Arras (2016). "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5512–5519. DOI: 10.1109/ICRA.2016.7487766 (cit. on p. 87).

Liu, Feng, Jan Sparbert, and Christoph Stiller (2008). "IMMPDA vehicle tracking system using asynchronous sensor fusion of radar and vision". In: *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 168–173. DOI: 10.1109/IVS.2008.4621161 (cit. on p. 26).

Llinas, James and David Lee Hall (1998). "An introduction to multi-sensor data fusion". In: *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*. Vol. 6, 537–540 vol.6. DOI: 10.1109/ISCAS.1998.705329 (cit. on p. 23).

Lucas, Bruce D. and Takeo Kanade (1981). "An Iterative Image Registration Technique with an Application to Stereo Vision". In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'81. Vancouver, BC, Canada: Morgan Kaufmann Publishers Inc., pp. 674–679. URL: http://dl.acm.org/citation.cfm?id=1623264.1623280 (cit. on p. 43).

Luettel, Thorsten, Michael Himmelsbach, and Hans-Joachim Wuensche (2012). "Autonomous Ground Vehicles 2014;Concepts and a Path to the Future". In: *Proceedings of the IEEE* 100.Special Centennial Issue, pp. 1831–1839. ISSN: 0018-9219. DOI: 10.1109/JPROC.2012.2189803 (cit. on p. 27).

Luo, Ren C., Chih-Chen Yih, and Kuo Lan Su (2002). "Multisensor fusion and integration: approaches, applications, and future research directions". In: *IEEE Sensors Journal* 2.2, pp. 107–119. ISSN: 1530-437X. DOI: 10.1109/JSEN.2002.1000251 (cit. on p. 32).

Ma, Bing (2001). "Parametric And Nonparametric Approaches For Multisensor Data Fusion". PhD thesis. The University of Michigan (cit. on p. 37).

Maggio, Emilio and Andrea Cavallaro (2009). "Learning Scene Context for Multiple Object Tracking". In: *IEEE Transactions on Image Processing* 18.8, pp. 1873–1884. ISSN: 1057-7149. DOI: 10.1109/TIP.2009.2019934 (cit. on p. 75).

Maggio, Emilio, Elisa Piccardo, Carlo Regazzoni, and Andrea Cavallaro (2007). "Particle PHD Filtering for Multi-Target Visual Tracking". In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. Vol. 1, pages. DOI: 10.1109/ICASSP.2007.366104 (cit. on p. 48).

Manohar, Vasant, Padmanabhan Soundararajan, Harish Raju, Dmitry Goldgof, Rangachar Kasturi, and John Garofolo (2006). "Performance Evaluation of Object Detection and Tracking in Video". In: *Computer Vision – ACCV 2006: 7th Asian Conference on Computer Vision, Hyderabad, India, January 13-16, 2006. Proceedings, Part II*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 151–161. ISBN: 978-3-540-32432-4. DOI: 10.1007/11612704_16. URL: http://dx.doi.org/10.1007/11612704_16 (cit. on p. 54).

Markovic, Ivan and Ivan Petrovic (2014). "Bayesian Sensor Fusion Methods for Dynamic Object Tracking - A Comparative Study". In: *Automatika – Journal for Control, Measurement, Electronics, Computing and Communications*. Vol. 55, pp. 386–398 (cit. on p. 37).

McKeever, Susan and Juan Ye (2013). "A Comparison of Evidence Fusion Rules for Situation Recognition in Sensor-Based Environments". In: *Evolving Ambient Intelligence: AmI 2013 Workshops, Dublin, Ireland, December 3-5, 2013. Revised Selected Papers*. Cham: Springer International Publishing, pp. 163–175. ISBN: 978-3-319-04406-4. DOI: 10.1007/978-3-319-04406-4_16. URL: http://dx.doi.org/10.1007/978-3-319-04406-4_16 (cit. on p. 38).

Meinhold, Richard J. and Nozer D.Singpurwalla (1983). "Understanding the Kalman Filter". In: *The American Statistician* 37 (cit. on p. 44).

Menze, Moritz and Andreas Geiger (2015). "Object scene flow for autonomous vehicles". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3061–3070. DOI: 10.1109/CVPR.2015.7298925 (cit. on p. 86).

Miller, Jeffrey (2008). "Vehicle-to-vehicle-to-infrastructure (V2V2I) intelligent transportation system architecture". In: *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 715–720. DOI: 10.1109/IVS.2008.4621301 (cit. on p. 32).

Molloy, Derek (2016). *Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux*. Wiley. ISBN: 978-1-119-1868-1. URL: http://www.exploringrpi.com/ (cit. on p. 89).

Mourllion, Benjamin, Dominique Gruyer, Cyril Royere, and Sébastien Theroude (2005). "Multi-hypotheses tracking algorithm based on the belief theory". In: *2005 7th International Conference on Information Fusion*. Vol. 2, pages. DOI: 10.1109/ICIF.2005.1591957 (cit. on p. 38).

Mutlu, Sinan, Tao Hu, and Oswald Lanz (2013). "Learning the Scene Illumination for Color-Based People Tracking in Dynamic Environment". In: *Image Analysis and Processing – ICIAP 2013: 17th International Conference, Naples, Italy, September 9-13, 2013, Proceedings, Part II*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 683–692. ISBN: 978-3-642-41184-7. DOI: 10.1007/978-3-642-41184-7_69. URL: http://dx.doi.org/10.1007/978-3-642-41184-7_69 (cit. on p. 85).

Napier, Ashley, Peter Corke, and Paul Newman (2013). "Cross-calibration of push-broom 2D LIDARs and cameras in natural scenes". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 3679–3684. DOI: 10.1109/ICRA.2013.6631094 (cit. on p. 39).

Neagoe, Victor and Cristian Tudoran (2008). "Road following for autonomous vehicle navigation using a concurrent neural classifier". In: *2008 World Automation Congress*, pp. 1–6 (cit. on p. 61).

Ng, Ka Ki and Edward J. Delp (2009). "New models for real-time tracking using particle filtering". In: *Visual Communications and Image Processing 2009*. Vol. 7257, pages. DOI: 10.1117/12.807311. URL: http://dx.doi.org/10.1117/12.807311 (cit. on p. 47).

Nobili, Simona, Salvador Dominguez, Gaetan Garcia, and Philippe Martinet (2015). "16 channels Velodyne versus planar LiDARs based perception system for Large Scale 2D-SLAM". In: (cit. on p. 27).

Oceanic, National and Atmospheric Administration (NOAA) Coastal Services Center. (2012). *Lidar 101: An Introduction to Lidar Technology, Data, and Applications*. URL: https://coast.noaa.gov/data/digitalcoast/pdf/lidar-101.pdf (cit. on p. 26).

Ofli, Ferda, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy (2013). "Berkeley MHAD: A comprehensive Multimodal Human Action Database". In: *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pp. 53–60. DOI: 10.1109/WACV.2013.6474999 (cit. on p. 86).

Olson, Edwin (2004). *A Primer on Odometry and Motor Control*. URL: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-186-mobile-autonomous-systems-laboratory-january-iap-2005/study-materials/odomtutorial.pdf (cit. on p. 90).

Orguner, Umut, Thomas B. Schon, and Fredrik Gustafsson (2009). "Improved target tracking with road network information". In: *2009 IEEE Aerospace conference*, pp. 1–11. DOI: 10.1109/AERO.2009.4839490 (cit. on p. 75).

Pandey, Gaurav, James R. McBride, Silvio Savarese, and Ryan M. Eustice (2014). "Automatic Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information". In: *Journal of Field Robotics*. ISSN: 1556-4967. DOI: 10.1002/rob.21542. URL: http://dx.doi.org/10.1002/rob.21542 (cit. on p. 39).

Pangop, Laurence Ngako, Frederic Chausse, Roland Chapuis, and Sebastien Cornou (2008). "Asynchronous Bayesian algorithm for object classification: Application to pedestrian detection in urban areas". In: *Information Fusion, 2008 11th International Conference on*, pp. 1–7 (cit. on p. 35).

Parekh, Himani S., Darshak G. Thakore, and Udesang K. Jaliya (2014). "A Survey on Object Detection and Tracking Methods". In: *International Journal of Innovative Research in Computer and Communication Engineering* 2 (cit. on p. 42).

Patino, Luis and James Ferryman (2014). "PETS 2014: Dataset and challenge". In: *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*, pp. 355–360. DOI: 10.1109/AVSS.2014.6918694 (cit. on p. 85).

Petrovskaya, Anna and Sebastian Thrun (2009). "Model based vehicle detection and tracking for autonomous urban driving". In: *Autonomous Robots* 26.2, pp. 123–139. ISSN: 1573-7527. DOI: 10.1007/s10514-009-9115-1. URL: http://dx.doi.org/10.1007/s10514-009-9115-1 (cit. on p. 17).

Pohl, Christine and John L. Van Genderen (1998). "Review article Multisensor image fusion in remote sensing: Concepts, methods and applications". In: *International Journal of Remote Sensing* 19.5, pp. 823–854. DOI: 10.1080/014311698215748. eprint: http://dx.doi.org/10.1080/014311698215748. URL: http://dx.doi.org/10.1080/014311698215748 (cit. on p. 34).

Pramanik, Sourav and Debotosh Bhattacharjee (2012). "Multi-sensor image fusion based on moment calculation". In: *Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on*, pp. 447–451. DOI: 10.1109/PDGC.2012.6449862 (cit. on p. 34).

Premebida, Cristiano, Goncalo Monteiro, Urbano Nunes, and Paulo Peixoto (2007). "A Lidar and Vision-based Approach for Pedestrian and Vehicle Detection and Tracking". In: *2007 IEEE Intelligent Transportation Systems Conference*, pp. 1044–1049. DOI: 10.1109/ITSC.2007.4357637 (cit. on pp. 27, 35).

Premebida, Cristiano and Urbano J. Carreira Nunes (2009). *Laser and Image Pedestrian Detection Dataset - LIPD*. URL: http://www2.isr.uc.pt/~cpremebida/dataset/ (cit. on p. 86).

Premebida, Cristiano, Paulo Peixoto, and Urbano Nunes (2006). "Tracking and Classification of Dynamic Obstacles Using Laser Range Finder and Vision". In: *IEEERSJ Internation Conference on Intelligent Robots and Systems*. IEEE (cit. on p. 35).

Quddus, Mohammed A., Washington Y. Ochieng, and Robert B. Noland (2007). "Current map-matching algorithms for transport applications: State-of-the art and future research directions". In: *Transportation Research Part C: Emerging Technologies* 15.5, pp. 312–328. ISSN: 0968-090X. DOI: http://dx.doi.org/10.1016/j.trc.2007.05.002. URL: http://www.sciencedirect.com/science/article/pii/S0968090X07000265 (cit. on p. 32).

Quigley et al. (2009). "ROS: an open-source Robot Operating System". In: *ICRA Workshop on Open Source Software* (cit. on p. 103).

Rezatofighi, Seyed Hamid, Anton Milan, Zhen Zhang, Qinfeng Shi, Anthony Dick, and Ian Reid (2015). "Joint Probabilistic Data Association Revisited". In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3047–3055. DOI: 10.1109/ICCV.2015.349 (cit. on p. 46).

Rodriguez, Sergio Alberto, Vincent Fremont, and Philippe Bonnifait (2008). "Influence of Intrinsic Parameters over Extrinsic Calibration between a Multi-Layer Lidar and a Camera". In: *IEEE 2nd Workshop on Planning, Perception and Navigation for Intelligent Vehicles*. Vol. 1, pp. 34–39 (cit. on p. 38).

Rodriguez, Sergio Alberto, Vincent Fremont, Philippe Bonnifait, and Veronique Cherfaoui (2011). "Multi-Modal Object Detection and Localization for High Integrity Driving Assistance". In: *Machine Vision Applications* 1, pp. 1–18. DOI: 10.1007/s00138-011-0386-0 (cit. on p. 38).

RxTools User Manual (2013). 1.10.0. Septentrio. URL: ftp://www.ngs.noaa.gov/pub/abilich/antcalCorbin/RxTools_Manual_v1.10.0.pdf (cit. on p. 94).

Sachs, Joachim, Michal Aftanas, James S. Crabbe, Milos Drutarovsky, Richard Klukas, Dusan Kocur, Trung-Thu Nguyen, Peter Peyerl, Jana Rovnakova, and Egor Zaikov (2008). "Detection and tracking of moving or trapped people hidden by obstacles using ultra-wideband pseudo-noise radar". In: *Radar Conference, 2008. EuRAD 2008. European*, pp. 408–411 (cit. on p. 23).

Salari, Vali and Ishwar Sethi (1990). "Feature Point Correspondence in the Presence of Occlusion". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 12.1, pp. 87–91. ISSN: 0162-8828. DOI: 10.1109/34.41387. URL: http://dx.doi.org/10.1109/34.41387 (cit. on p. 42).

Sankaranarayanan, Aswin C., Ashok Veeraraghavan, and Rama Chellappa (2008). "Object detection, tracking and recognition for multiple smart cameras". In: *Proceedings of the IEEE* 96.10, pp. 1606–1624 (cit. on p. 24).

Sato, Koichi and Jake K. Aggarwal (2004). "Temporal spatio-velocity transform and its application to tracking and interaction". In: *Computer Vision and Image Understanding* 96.2. Special Issue on Event Detection in Video, pp. 100–128. ISSN: 1077-3142. DOI: http://dx.doi.org/10.1016/j.cviu.2004.02.003. URL: http://www.sciencedirect.com/science/article/pii/S1077314204000700 (cit. on p. 43).

Sattarov, Egor, Sergio Alberto Rodríguez Florez, Alexander Gepperth, and Roger Reynaud (2014). "Context-based vector fields for multi-object tracking in application to road traf-

fic". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1179–1185. DOI: 10.1109/ITSC.2014.6957847 (cit. on pp. 52, 75).

Sattarov, Egor, Alexander Gepperth, Sergio Alberto Rodriguez Florez, and Roger Reynaud (2015). "Calibration-free match finding between vision and LIDAR". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1061–1067. DOI: 10.1109/IVS.2015.7225825 (cit. on p. 60).

Scaramuzza, Davide, Ahad Harati, and Roland Siegwart (2007). "Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes". In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 4164–4169. DOI: 10.1109/IROS.2007.4399276 (cit. on p. 39).

Schall, Gerhard, Daniel Wagner, Gerhard Reitmayr, Elise Taichmann, Manfred Wieser, Dieter Schmalstieg, and Bernhard Hofmann-Wellenhof (2009). "Global pose estimation using multi-sensor fusion for outdoor Augmented Reality". In: *ISMAR* (cit. on p. 31).

Scharwächter, Timo, Markus Enzweiler, Uwe Franke, and Stefan Roth (2013). "Efficient Multi-cue Scene Segmentation". In: *Pattern Recognition: 35th German Conference, GCPR 2013, Saarbrücken, Germany, September 3-6, 2013. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 435–445. ISBN: 978-3-642-40602-7. DOI: 10.1007/978-3-642-40602-7_46. URL: http://dx.doi.org/10.1007/978-3-642-40602-7_46 (cit. on p. 86).

— (2014). "Stixmantics: A Medium-Level Model for Real-Time Semantic Scene Understanding". In: *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*. Cham: Springer International Publishing, pp. 533–548. ISBN: 978-3-319-10602-1. DOI: 10.1007/978-3-319-10602-1_35. URL: http://dx.doi.org/10.1007/978-3-319-10602-1_35 (cit. on p. 86).

Scherzinger, Bruno M. (2000). *Precise Robust Positioning with Inertial/GPS RTK* (cit. on p. 31).

Schikora, Marek, Amadou Gning, Lyudmila Mihaylova, Daniel Cremers, and Wolfgang Koch (2014). "Box-particle probability hypothesis density filtering". In: *IEEE Trans. Aerospace and Electronic Systems* 50, pp. 1660–1672 (cit. on p. 48).

Shafer, Glenn (1976). *A Mathematical Theory of Evidence*. Princeton: Princeton University Press (cit. on p. 37).

Sharma, Kajal, Kwang-Young Jeong, and Sung-Gaun Kim (2011). "Vision based autonomous vehicle navigation with self-organizing map feature matching technique". In: *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*, pp. 946–949 (cit. on p. 62).

Shi, Jianbo and Carlo Tomasi (1994). "Good Features to Track". In: *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pp. 593–600 (cit. on p. 43).

Shibata, Naoki, Seiji Sugiyama, and Takahiro Wada (2014). "Collision avoidance control with steering using velocity potential field". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 438–443. DOI: 10.1109/IVS.2014.6856469 (cit. on p. 75).

Shimojo, Shinsuke and Ladan Shams (2001). "Sensory modalities are not separate modalities: plasticity and interactions". In: *Current Opinion in Neurobiology* (cit. on p. 22).

Smets, Philippe (1988). "Belief functions versus probability functions". In: *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, pp. 17–24 (cit. on p. 37).

— (1994). "Advances in the Dempster-Shafer Theory of Evidence". In: New York, NY, USA: John Wiley & Sons, Inc. Chap. What is Dempster-Shafer's Model?, pp. 5–34. ISBN: 0-471-55248-8. URL: http://dl.acm.org/citation.cfm?id=186965.186966 (cit. on pp. 37, 38).

Smith, Kevin, Daniel Gatica-Perez, Jean-Marc Odobez, and Sileye Ba (2005). "Evaluating Multi-Object Tracking". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pp. 36–36. DOI: 10.1109/CVPR.2005.453 (cit. on p. 54).

Stiller, Christoph, Fernando Puente León, and Marco Kruse (2011). "Information fusion for automotive applications – An overview". In: *Information Fusion* 12.4. Special Issue on Information Fusion for Cognitive Automobiles, pp. 244–252. ISSN: 1566-2535. DOI: http://dx.doi.org/10.1016/j.inffus.2011.03.005. URL: http://www.sciencedirect.com/science/article/pii/S1566253511000273 (cit. on p. 35).

Strygulec, Sarah, Dennis Müller, Mirko Meuter, Christian Nunn, Sharmila Ghosh, and Christian Wöhler (2013). "Road Boundary Detection and Tracking using monochrome camera images". In: *Information Fusion (FUSION), 2013 16th International Conference on*, pp. 864–870 (cit. on p. 75).

Sturm, Jürgen, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers (2012). "A Benchmark for the Evaluation of RGB-D SLAM Systems". In: *Proc. of the International Conference on Intelligent Robot Systems (IROS)* (cit. on p. 85).

Sun, Zehang, George Bebis, and Ronald Miller (2006). "On-road vehicle detection: a review". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.5, pp. 694–711. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2006.104 (cit. on p. 34).

Sunderrajan, Santhoshkumar and Bangalore S. Manjunath (2013). "Multiple view discriminative appearance modeling with IMCMC for distributed tracking". In: *Distributed Smart Cameras (ICDSC), 2013 Seventh International Conference on*, pp. 1–7. DOI: 10.1109/ICDSC.2013.6778203 (cit. on p. 85).

— (2016). "Context-Aware Hypergraph Modeling for Re-identification and Summarization". In: *Multimedia, IEEE Transactions on* 18.1, pp. 51–63. ISSN: 1520-9210. DOI: 10.1109/TMM.2015.2496139 (cit. on p. 85).

Svoboda, Tomáš, Hanspeter Hug, and Luc Van Gool (2002). "ViRoom — Low Cost Synchronized Multicamera System and Its Self-calibration". In: *Pattern Recognition: 24th DAGM Symposium Zurich, Switzerland, September 16–18, 2002 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 515–522. ISBN: 978-3-540-45783-1. DOI: 10.1007/3-540-45783-6_62. URL: http://dx.doi.org/10.1007/3-540-45783-6_62 (cit. on p. 24).

Taj, Murtaza, Emilio Maggio, and Andrea Cavallaro (2008). "Objective Evaluation of Pedestrian and Vehicle Tracking on the CLEAR Surveillance Dataset". In: *Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR 2007 and RT 2007, Baltimore, MD, USA, May 8-11, 2007, Revised Selected Papers*. Berlin, Heidelberg:

Springer Berlin Heidelberg, pp. 160–173. ISBN: 978-3-540-68585-2. DOI: 10.1007/978-3-540-68585-2_13. URL: http://dx.doi.org/10.1007/978-3-540-68585-2_13 (cit. on p. 86).

Takasu, Tomoji (2013). *RTKLIB ver. 2.4.2 Manual*. URL: http://www.rtklib.com/prog/manual_2.4.2.pdf (cit. on p. 94).

Tao, Hai, Harpreet S. Sawhney, and Rakesh Kumar (2002). "Object tracking with Bayesian estimation of dynamic layer representations". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.1, pp. 75–89. ISSN: 0162-8828. DOI: 10.1109/34.982885 (cit. on p. 42).

Toledo-Moreo, Rafael, Miguel A. Zamora-Izquierdo, Benito Ubeda-Minarro, and Antonio F. Gomez-Skarmeta (2007). "High-Integrity IMM-EKF-Based Road Vehicle Navigation With Low-Cost GPS/SBAS/INS". In: *IEEE Transactions on Intelligent Transportation Systems* 8.3, pp. 491–511. ISSN: 1524-9050. DOI: 10.1109/TITS.2007.902642 (cit. on p. 30).

u-blox 8 / u-blox M8 Receiver Description (2016). R11. U-blox. URL: https://www.u-blox.com/sites/default/files/products/documents/u-blox8-M8_ReceiverDescrProtSpec_(UBX-13003221)_Public.pdf (cit. on p. 93).

Ulmke, Martin and Karl W. Koch (2006). "Road Map Extraction using GMTI Tracking". In: *2006 9th International Conference on Information Fusion*, pp. 1–7. DOI: 10.1109/ICIF.2006.301564 (cit. on p. 75).

Urmson, Christopher et al. (2009). "Autonomous Driving in Urban Environments: Boss and the Urban Challenge". In: *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–59. ISBN: 978-3-642-03991-1. DOI: 10.1007/978-3-642-03991-1_1. URL: http://dx.doi.org/10.1007/978-3-642-03991-1_1 (cit. on pp. 23, 26).

Vasic, Milos and Alcherio Martinoli (2015). "A Collaborative Sensor Fusion Algorithm for Multi-object Tracking Using a Gaussian Mixture Probability Hypothesis Density Filter". In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 491–498. DOI: 10.1109/ITSC.2015.87 (cit. on p. 36).

Vatavu, Andrei, Arthur D. Costea, and Sergiu Nedevschi (2015). "Modeling and tracking of dynamic obstacles for logistic plants using omnidirectional stereo vision". In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 3552–3558. DOI: 10.1109/IROS.2015.7353873 (cit. on p. 24).

Vatavu, Andrei, Radu Danescu, and Sergiu Nedevschi (2015). "Stereovision-Based Multiple Object Tracking in Traffic Scenarios Using Free-Form Obstacle Delimiters and Particle Filters". In: *IEEE Transactions on Intelligent Transportation Systems* 16.1, pp. 498–511. ISSN: 1524-9050. DOI: 10.1109/TITS.2014.2366248 (cit. on p. 25).

Veenman, Cor J., Marcel J.T. Reinders, and Eric Backer (2001). "Resolving motion correspondence for densely moving points". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.1, pp. 54–72. ISSN: 0162-8828. DOI: 10.1109/34.899946 (cit. on p. 42).

Vislab (2011). *Multisensor Datasets*. URL: http://vislab.it/products/multisensor-datasets/ (cit. on p. 87).

Vo, Ba-Ngu and Wing-Kin Ma (2006). "The Gaussian Mixture Probability Hypothesis Density Filter". In: *IEEE Transactions on Signal Processing* 54.11, pp. 4091–4104. ISSN: 1053-587X. DOI: 10.1109/TSP.2006.881190 (cit. on p. 48).

Vo, Ba-Ngu, S. Singh, and A. Doucet (2003). "Sequential monte carlo implementation of the phd filter for multi-target tracking". In: *Information Fusion, 2003. Proceedings of the Sixth International Conference of*. Vol. 2, pp. 792–799. DOI: 10.1109/ICIF.2003.177320 (cit. on p. 48).

Vu, Trung-Dung, Olivier Aycard, and Fabio Tango (2014). "Object perception for intelligent vehicle applications: A multi-sensor fusion approach". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 774–780. DOI: 10.1109/IVS.2014.6856588 (cit. on p. 35).

Wang, Meng, Yaping Dai, Yan Liu, and Tian Yanbing (2010). "Feature-level image sequence fusion based on histograms of Oriented Gradients". In: *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*. Vol. 9, pp. 265–269. DOI: 10.1109/ICCSIT.2010.5563759 (cit. on p. 35).

Wang, Zhiwen, Shaozi Li, Qixian Cai, Songzhi Su, and MeiZhen Liu (2009). "Multi-spectrum image fusion algorithm based on weighted and improved wavelet transform". In: *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*. Vol. 4, pp. 63–66. DOI: 10.1109/ICICISYS.2009.5357741 (cit. on p. 34).

Welch, Greg and Eric Foxlin (2002). "Motion tracking: no silver bullet, but a respectable arsenal". In: *IEEE Computer Graphics and Applications* 22.6, pp. 24–38. ISSN: 0272-1716. DOI: 10.1109/MCG.2002.1046626 (cit. on p. 23).

Wen, Xiaofei (2011). "Image Fusion Based on Improved IHS Transform with Weighted Average". In: *Computational and Information Sciences (ICCIS), 2011 International Conference on*, pp. 111–113. DOI: 10.1109/ICCIS.2011.162 (cit. on p. 34).

World Geodetic System — 1984 (WGS - 84) Manual (2002). 2nd ed. International Civil Aviation Organization. URL: http://www.icao.int/NACC/Documents/Meetings/2014/ECARAIM/REF08-Doc9674.pdf (cit. on p. 98).

Wu, Huadong (2004). "Sensor Data Fusion for Context-aware Computing Using Dempster-shafer Theory". AAI3126933. PhD thesis. Pittsburgh, PA, USA: Carnegie Mellon University (cit. on p. 37).

Wu, Zheng, Nathan Fuller, Diane Theriault, and Margrit Betke (2014). "A Thermal Infrared Video Benchmark for Visual Analysis". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 201–208. DOI: 10.1109/CVPRW.2014.39 (cit. on p. 86).

Yilmaz, Alper, Omar Javed, and Mubarak Shah (2006). "Object Tracking: A Survey". In: *ACM Comput. Surv.* 38.4. ISSN: 0360-0300. DOI: 10.1145/1177352.1177355. URL: http://doi.acm.org/10.1145/1177352.1177355 (cit. on p. 42).

Yin, Fei, Dimitrios Makris, and Sergio A. Velastin (2007). "Performance evaluation of object tracking algorithms". In: *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Rio De Janeiro, Brazil*. Citeseer, p. 25 (cit. on p. 54).

Zhang, Zhengyou (1999). "Flexible camera calibration by viewing a plane from unknown orientations". In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International*

Conference on. Vol. 1, 666–673 vol.1. DOI: 10.1109/ICCV.1999.791289 (cit. on pp. 39, 95, 96).

Zheng, Sicong (2010). "Pixel-level Image Fusion Algorithms for Multi-camera Imaging System". MA thesis. University of Tennessee, Knoxville (cit. on p. 34).

Zheng, Yunmei, Zhiguo Shi, Rongxing Lu, Shaohua Hong, and Xuemin Shen (2013). "An Efficient Data-Driven Particle PHD Filter for Multitarget Tracking". In: *IEEE Transactions on Industrial Informatics* 9.4, pp. 2318–2326. ISSN: 1551-3203. DOI: 10.1109/TII.2012.2228875 (cit. on p. 48).

Synthèse substantiel

Aujourd'hui, la conduite autonome et les véhicules intelligents sont des applications qui comportent de verrous scientifiques et technologiques importants. Parmi ses verrous, nous portons un intérêt particulier aux problèmes liés à la localisation, l'analyse et la compréhension de scène dynamique et aux interactions d'un véhicule autonome avec d'autres véhicules, avec l'infrastructure et avec les usagers vulnérables de la route (e.g. vélos et piétons).

Problèmes étudiées

Dans le cadre de l'analyse de scène dynamique, l'étude des méthodes de suivi multi-objets basées sur la perception embarquée est d'une grande importance. Le suivi d'objets dynamiques est un problème complexe dû principalement aux spécificités et à la multiplicité de contraintes des environnements observés (rural, semi-urbain, urbain). De plus, le méthodes doivent aussi s'adapter aux limitations des capteurs (imprécision, fiabilité) et aux ressources limitées à bord du véhicule.

Le suivi basé mono-capteur est un problème non trivial; le suivi basé multi-capteurs est une tâche encore plus sophistiquée car l'information des capteurs différents doit être fusionnée en prenant en compte leurs imprécisions intrinsèques. La fusion des données a donc pour but l'augmentation de la précision et de la confiance des estimations issues du suivi. C'est-à-dire, la combinaison des capteurs multiples doit rajouter de l'inférence plus spécifique qu'un seul capteur. Les approches multi-capteurs s'adaptent bien aux véhicules autonomes puisqu'ils sont équipés de caméras, de RADARs, LIDARs, de GPS, de systèmes inertiels, etc.

Tandis que la fusion des données enrichie le processus de suivi grâce à la combinaison des données observées (redondance et/ou complémentarité), une autre approche consiste à utiliser de l'information additionnelle dite contextuelle, afin d'améliorer le suivi. L'information du contexte peut être extraite des observations brutes, comme la détection des voies de circulation par vision ou de manière décorrélée du processus de perception comme à partir de systèmes d'information géographiques (SIG).

Le suivi multi-objets fournit de l'information clé pour l'évitement d'obstacles, alors, il doit être précis, continu et intègre. Ces critères de qualité sont impactés négativement quand les objets sont détectés partialement ou même complètement occlus. Les techniques de fusion de l'état de l'art combinent l'information des sources multiples afin d'élargir le champ de vision en utilisant des capteurs avec une portée différente. D'autres stratégies de fusion diminuent les occlusions et les non détections. Toutes ces méthodes sont limitées par des erreurs intrinsèques des capteurs, par des hypothèses sur les modèles employés, par la complexité des données à associer et par l'incertitude des paramètres du système.

La motivation de ce travail supportée par le fait que la complexité de l'environnement peut être réduite quand l'information contextuelle est prise en compte par le système de perception. Ce type d'information est souvent disponible sous une représentation sémantique. Les questions qu'on adresse sont : Comment intégrer l'information contextuelle d'une manière efficace? Comment intégrer la prise en compte de ce type d'information? Comment concevoir un algorithme robuste pour évaluer toutes les données même en cas de l'incomplétude?

Après d'avoir présenté la méthodologie, nous devons aussi être capable d'évaluer et de comparer les solutions proposées. Pour ce faire, une base de données composée des enregistrements des capteurs accompagnés par les annotations de Vérité de Terrain a été créée.

Finalement, l'investigation des problèmes instanciés n'a pas seulement adressé les tâches annoncées, mais aussi retrouvé de nouvelles perspectives vers un système pour le suivi des objets multi-modal assisté par des informations contextuelles. Cette idée peut conduire à la généralisation du système implanté vers un formalisme mathématique unique.

État de l'art

Dans le chapitre 1, l'état de l'art concernant le suivi multi-objet multimodal est décrit et détaillé. Il présente les techniques appliquées aux systèmes de perception pour les véhicules autonomes telles que les caméras et leurs différentes variantes, le RADAR, le LIDARet les caméras TOF. Les méthodes de positionnement avec les systèmes GNSS (i.e. GPS) et leurs augmentations de précision sont expliquées en détail afin de faciliter la compréhension du chapitre 5. Les systèmes qui prennent en compte les sources de l'information extérieure, du type V2V, V2I ont été aussi considérés. La section *Méthodologie* comprend la classification des méthodes de fusion des données suivant différents niveaux d'abstraction de la représentation de données. La présentation de l'état de l'art des mécanismes d'association entre les données multi-capteur ainsi que les méthodes d'apprentissage et de calibration multi-capteur clôturent ce chapitre.

Le suivi des objets dynamiques

Le chapitre 2 explique les définitions fondamentales et il spécifie le processus récursive de suivi des objets. Dans la première section, l'état de l'art des approches de suivi est présenté. Le modèle d'objet utilisé est défini et formalisé. Ensuite, un accent est réservé aux méthodes de suivi probabilistes. Dans la deuxième section, un système pour le suivi des objets multiples est proposé. Ce dernier système est basé sur les méthodes existantes. Le système est développé et implanté. Le système de suivi d'objets se base sur un formalise bayésien dans une implémentation de type Monte-Carlo. Cette représentation permet l'intégration de l'information contextuelle (voir le chapitre 4) peu complexe au niveau des particules. L'efficacité du suivi des objets multiples est quantifiée à la fin de chapitre.

Association de données multi-capteurs

Le chapitre 3 s'adresse à l'association de données multi-capteurs. Une méthode probabiliste d'apprentissage pour l'association entre les espaces multi-capteurs a été étudiée et proposée.

L'avantage principale de l'approche est l'absence du besoin d'une procédure de calibration avec une grande précision adapté aux applications intégrant LIDAR-vison. Ce chapitre constitue une contribution pour le suivi multimodal. Il est important de noter que le terme *association* est largement utilisé en suivi comme l'association temporelle entre les détections d'un objet. Ici, l'association concerne exclusivement l'association spatiale entre deux espaces de capteurs. La méthode proposée est générique et peut gérer l'association entre les espaces quand elle n'est pas unique.

L'approche est basée sur deux points principaux: 1) La composition des cartes auto-organisatrices (SOM) modélise la densité des détections pour chaque capteur. 2) Le cumul des statistiques des paires des détections observées et ses attributions aux nœuds de SOMs.

L'évaluation de la méthode d'association a été faite sur de bases de données publiques.

Intégration de l'information contextuelle dans le suivi multi-objets

Le chapitre 4 décrit la méthode proposée pour l'intégration de l'information contextuelle dans le processus du suivi pour améliorer la précision et la certitude. L'approche est basée sur la représentation probabiliste du système. L'avantage apporté par cette implémentation est la possibilité de déterminer si l'objet suivi décrit un comportement attendu ou inattendu. Cette information est essentielle pour les applications du type sécurité pour les véhicules intelligents. Les expériences démontrent l'amélioration de suivi d'objet qui respecte l'information contextuelle et l'absence de la dégradation considérable quand l'objet suivi ne respecte pas les informations contextuelles. En détail, le schéma d'intégration du contexte dans le filtre de particules consiste à replacer partiellement les particules qui représentent l'état de l'objet par les particules qui sont définies par les informations contextuelles. Le contexte dans les cas évalués a été représenté sous la forme des annotations des cartes "Open Street Maps".

Base de données

Le chapitre 4 décrit un protocole détaillé pour la création d'une base de données en utilisant les enregistrements des capteurs multiples intégrés à bord d'un véhicule intelligent et d'autres rattachées aux objets à suivre. L'étude des bases de données existantes est présentée remarquant le besoin d'une base de données avec des annotations de Vérité Terrain multicapteur en applications d'extérieur. L'avantage principale de cette base de données est l'indépendance des informations du type Vérité Terrain par rapport aux autres capteurs embarqués. La calibration précise des capteurs, Vérité Terrain et les observations brutes sont incluses dans la base de données. Les enregistrements brutes sont asynchrones et comportent la vision monoculaire, le LIDAR, des odomètres et un système de positionnement GPS RTK.

Résultats

Le chapitre 6 rapporte l'évaluation de toutes les méthodes proposées en utilisant la base de données enregistrée. Les résultats confirment les conclusions de la performance des méthodes dans de scénarios réels.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Etude et quantification de la contribution des systèmes de perception multimodale assistés par des informations de contexte pour la détection et le suivi d'objets dynamiques

Mot-clefs: Fusion des données, perception, suivi, multimodalité

Résumé: Cette thèse a pour but d'étudier et de quantifier la contribution de la perception multimodale assistée par le contexte pour suivre des objets en mouvement. Cette étude sera appliquée à la reconnaissance des objets pertinents dans les environnements de la circulation pour les véhicules intelligents (VI). Les résultats à obtenir devront permettre de transposer le concept proposé à un ensemble plus large de capteurs et de classes d'objets en utilisant une approche système intégrative qui implique des méthodes d'apprentissage. En particulier, ces méthodes d'apprentissage vont examiner comment l'implantation dans un système intégré, qui prévoie une multitude des sources de données différentes, peut conduire à apprendre 1) sans ou avec une supervision limitée, réduite en exploitant des corrélations 2) de façon incrémentale à la connaissance stockée au lieu de faire un entraînement complet à chaque fois qu'une nouvelle donnée arrive 3) collectivement à chaque instant d'apprentissage dans le système entraîné d'une manière qui assure approximativement une fusion optimale. Concrètement, le couplage fort entre les classifiers des objets en modalités multiples aussi bien que l'extraction du contexte de la géométrie de la scène sont à étudier: d'abord en théorie, après en application du trafic routier. La nouveauté de l'approche d'intégration envisagée se pose dans le couplage fort entre les composants du système, tels que la segmentation, le suivi des objets, l'estimation de la géométrie de la scène et la catégorisation des objets basée sur la stratégie de l'inference probabiliste. Une telle stratégie caractérise des systèmes où toutes les composants de perception émettent et reçoivent les distributions des résultats possibles avec leur score de croyance probabiliste attribué. De cette façon, chaque composant de traitement peut prendre en compte les résultats des autres composants au niveau plus bas par rapport aux combinaisons des résultats finaux. Cela diminue beaucoup le temps et les ressources pour le calcul, quand les techniques de l'application de l'inference Bayésienne garantissent que les données d'entrée peu plausible n'apportent pas des impactes négatives.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91,190 Saint-Aubin, France

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Contributions of context-aided multi-modal perception systems for detection and tracking of moving objects

Key words: Multisensor data fusion, perception, tracking, multi-modal

Abstract: This thesis project will investigate and quantify the contribution of context-aided multimodal perception for tracking moving objects. This research study will be applied to the recognition of relevant objects in road traffic environments for Intelligent Vehicles (IV). The results to be obtained will allow us to transpose the proposed concept to a wide range of state-of-the-art sensors and object classes by means of an integrative system approach involving learning methods. In particular, such learning methods will investigate how the embedding into an embodied system providing a multitude of different data sources, can be harnessed to learn 1) without, or with reduced, explicit supervision by exploiting correlations 2) incrementally, by adding to existing knowledge instead of complete retraining every time new data arrive 3) collectively, each learning instance in the system being trained in a way that ensures approximately optimal fusion. Concretely, a tight coupling between object classifiers in multiple modalities as well as geometric scene context extraction will be studied, first in theory, then in the context of road traffic. The novelty of the envisioned integration approach lies in the tight coupling between system components such as object segmentation, object tracking, scene geometry estimation and object categorization based on a probabilistic inference strategy. Such a strategy characterizes systems where all perception components broadcast and receive distributions of multiple possible results together with a probabilistic belief score. In this way, each processing component can take into account the results of other components at a much earlier stage (as compared to just combining final results), thus hugely increasing its computation power, while the application of Bayesian inference techniques will ensure that implausible inputs do not cause negative effects.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91,190 Saint-Aubin, France