Contributions by Vision Systems to Multi-sensor Object Localization and Tracking for Intelligent Vehicles

Sergio Alberto Rodriguez Florez

University of Technology of Compiègne

Thesis

Submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Area of specialization: Information Technologies and Systems

by

Sergio Alberto RODRÍGUEZ FLÓREZ

Contributions by Vision Systems to Multi-sensor Object Localization and Tracking for Intelligent Vehicles

Heudiasyc Laboratory, UMR UTC/CNRS 6599

Defended on December 7th, 2010.

Thesis Committee:

Reviewers:	Christoph Stiller	Karlsruhe Institute of Technology-Germany
	Patrick Rives	INRIA Sophia Antipolis
Examiners:	Véronique Cherfaoui	UTC-Compiègne
	Urbano Nunes	University of Coimbra-Portugal
	Didier Aubert	INRETS
	Javier Ibañez-Guzmán	RENAULT
Supervisors:	Vincent Frémont	UTC-Compiègne
	Philippe Bonnifait	UTC-Compiègne

Université de Technologie de Compiègne

Thèse

pour obtenir le grade de

Docteur

Spécialité: Technologie de l'Information et de Systems (TIS)

par

Sergio Alberto RODRÍGUEZ FLÓREZ

Contributions des systèmes de vision à la localisation et au suivi d'objets par fusion multi-capteur pour les véhicules intelligents

Laboratoire Heudiasyc, Unité Mixte de Recherche UTC/CNRS 6599

Soutenue le 7 Decembre, 2010 devant le jury constitué de :

Rapporteurs:	Christoph Stiller	Karlsruhe Institute of Technology-Germany
	Patrick Rives	INRIA Sophia Antipolis
Examinateurs:	Véronique Cherfaoui	UTC-Compiègne
	Urbano Nunes	University of Coimbra-Portugal
	Didier Aubert	INRETS
	Javier Ibañez-Guzmán	RENAULT
Directeurs de thèse:	Vincent Frémont	UTC-Compiègne
	Philippe Bonnifait	UTC-Compiègne

Acknowledgments

This PhD thesis was carried out at the Heudiasyc Laboratory in the University of Technology of Compiègne from October 2007 to December 2010. It was financed with a grant from the French National Education and Research Ministry. All this work would not have been possible without the help of many people. Here, I just want to express my gratitude to all those people who supported and contributed to this thesis.

First, I wish to express my sincere thanks to my supervisors Vincent Frémont and Prof. Philippe Bonnifait, for their valuable advices, their continuous encouragements and their personal involvement. I would like to thank Prof. Jean Giroire for his helpful discussions and for sharing his knowledge with me. I would like to express my gratitude to Gérald Derhbomez and Thierry Monglon for their investment during the experimental tests that were accomplished during this study. Finally, I want to thank my parents for their unconditional and constant support, and thanks to Dima, for her love, support and patience.

Abstract

Advanced Driver Assistance Systems (ADAS) can improve road safety by supporting the driver through warnings in hazardous circumstances or triggering appropriate actions when facing imminent collision situations (e.g. airbags, emergency brake systems, etc). In this context, the knowledge of the location and the speed of the surrounding mobile objects constitute a key information.

Consequently, in this work, we focus on object detection, localization and tracking in dynamic scenes. Noticing the increasing presence of embedded multi-camera systems on vehicles and recognizing the effectiveness of lidar automotive systems to detect obstacles, we investigate stereo vision systems contributions to multi-modal perception of the environment geometry.

In order to fuse geometrical information between lidar and vision system, we propose a calibration process which determines the extrinsic parameters between the exteroceptive sensors and quantifies the uncertainties of this estimation.

We present a real-time visual odometry method which estimates the vehicle ego-motion and simplifies dynamic object motion analysis.

Then, the integrity of the lidar-based object detection and tracking is increased by the means of a visual confirmation method that exploits stereo-vision 3D dense reconstruction in focused areas.

Finally, a complete full scale automotive system integrating the considered perception modalities was implemented and tested experimentally in open road situations with an experimental car.

Résumé

Les systèmes d'aide à la conduite peuvent améliorer la sécurité routière en aidant les utilisateurs via des avertissements de situations dangereuses ou en déclenchant des actions appropriées en cas de collision imminente (airbags, freinage d'urgence, etc). Dans ce cas, la connaissance de la position et de la vitesse des objets mobiles alentours constitue une information clé.

C'est pourquoi, dans ce travail, nous nous focalisons sur la détection et le suivi d'objets dans une scène dynamique. En remarquant que les systèmes multi-caméras sont de plus en plus présents dans les véhicules et en sachant que le lidar est performant pour la détection d'obstacles, nous nous intéressons à l'apport de la vision stéréoscopique dans la perception géométrique multimodale de l'environnement.

Afin de fusionner les informations géométriques entre le lidar et le système de vision, nous avons développé un procédé de calibrage qui détermine les paramètres extrinsèques et évalue les incertitudes sur ces estimations.

Nous proposons ensuite une méthode d'odométrie visuelle temps-réel permettant d'estimer le mouvement propre du véhicule afin de simplifier l'analyse du mouvement des objets dynamiques.

Dans un second temps, nous montrons comment l'intégrité de la détection et du suivi des objets par lidar peut être améliorée en utilisant une méthode de confirmation visuelle qui procède par reconstruction dense de l'environnement 3D.

Pour finir, le système de perception multimodal a été intégré sur une plateforme automobile, ce qui a permis de tester expérimentalement les différentes approches proposées dans des situations routières en environnement non contrôlé.

Contents

List of Symbols

Sets, Linear Spaces and Geometry

$\mathbb{R}$, $\mathbb{C}$	Sets of real and complex numbers respectively
$\mathbb{R}^n$, $\mathbb{C}^n$	Sets of n-dimensional real and complex linear space
$\mathbb{H}$	Hamiltonian space or Quaternion set
$\mathbb{E}^3$	Three-dimensional Euclidean space
p, x	A generic point in the n-dimensional real space $\mathbb{R}^n$ (abstract)
$\vec{v}$	A generic vector in the n-dimensional real space $\mathbb{R}^n$ (abstract)
L	A generic 1-D line in the space
Ω	A generic 2-D plane in the space
$\mathbf{p}$	Coordinates $\mathbf{p} \in \mathbb{R}^n$ of a generic point p
$\mathbf{x}$	An n-dimensional vector
$\mathbf{X}$	A matrix
$\mathbf{x}^T$, $\mathbf{X}^T$	Transpose of $\mathbf{x}$; of $\mathbf{X}$
$\mathbf{X}^\dagger$	Pseudo-inverse matrix
$f(\cdot)$	A scalar function
$f(x, y)$	A multivariate function
$\mathbf{f}(\cdot)$	A vector function
$\dot{\mathbf{f}}$	Time-derivative of the function $\mathbf{f}$
$\mathbf{F}$	The Jacobian matrix
$\mathbf{F}_\mathbf{x}$	The Jacobian matrix of $\mathbf{f}$ with respect to $\mathbf{x}$
$\mathbf{H}$	The Hessian matrix
α, τ	Scalar tuning parameters
ϕ, β, ψ	Angle variables
$\times$	Vectorial product operator
$\mathbf{U}\,\mathbf{S}\,\mathbf{V}^T$	Singular Value Decomposition (SVD) of a matrix

Rigid Transformations and Frames

$\mathbf{q}$	Quaternion rotation form $\mathbf{q} = q_0 + q_1 i + q_2 j + q_3 ij$
$\mathbf{r} = [\phi,\, \theta,\, \psi]^T$	Euler rotation form (Euler angles)
$\mathbf{R}$, $\mathbf{t}$	Rotation matrix and translation vector
$\mathbf{p}_{(x,y,z)} = [\mathbf{p}_x,\, \mathbf{p}_y,\, \mathbf{p}_z]^T$	The metric coordinate components of a 3D point $\mathbf{p}$
$\bar{\mathbf{q}}$	Conjugate of the quaternion $\mathbf{q}$
$\bar{\mathbf{q}}_1 \star \bar{\mathbf{q}}_2$	Quaternion product
$\underline{\mathbf{p}}$	Expanded vector of $\mathbf{p}$ corresponding to $\underline{\mathbf{p}} = \begin{bmatrix} 0 & \mathbf{p}^T \end{bmatrix}^T$
$\mathbf{q}(\mathbf{R})$	Quaternion representation of the rotation matrix $\mathbf{R}$
$\mathbb{I}_{n\times n}$	An n-dimensional identity matrix
$\mathcal{F}$	A reference frame
$\mathcal{F}\{RDF\}$	Orientation of a reference frame (e.g. X-Right, Y-Down, Z-Front)
$\mathcal{F}(t)$	A time-dependent frame (mobile frame)
${}^{\mathcal{F}}\mathbf{p}$	The coordinates of a point p defined in the frame $\mathcal{F}$
${}^{\mathcal{F}(t)}\mathbf{p}$	The coordinates of a point p defined in the mobile frame $\mathcal{F}(t)$
${}^{\mathcal{G}}\mathcal{F}$	A $\mathcal{F}$ frame defined with respect to a $\mathcal{G}$ Frame
${}^{\mathcal{G}}[\mathbf{R},\, \mathbf{t}]_{\mathcal{F}}$	A Rigid-body transformation from $\mathcal{F}$ to $\mathcal{G}$ frame
${}^{\mathcal{G}}[\mathbf{q},\, \mathbf{t}]_{\mathcal{F}}$	A frame transformation from $\mathcal{F}$ to $\mathcal{G}$ frame (Quaternion)
${}^{\mathcal{G}}[\boldsymbol{\omega},\, \mathbf{t}]_{\mathcal{F}}$	A frame transformation from $\mathcal{F}$ to $\mathcal{G}$ frame (Axe-Angle Rotation)
${}^{LLH}\mathbf{p} = [\mathbf{p}_{(lat)},\, \mathbf{p}_{(lon)},\, \mathbf{p}_{(alt)}]^T$	A positioning point in the the World Geodetic System (WGS-84) frame
${}^{ENU}\mathbf{p}$	A 3D point defined in the The East-North-Up frame
${}^{ECEF}\mathbf{p}$	A 3D point defined in the Earth-Centered Earth-Fixed (ECEF) frame

Computer Vision

$\mathbf{K}$	Intrinsic parameters matrix
$f_x,\, f_y,\, u_0,\, v_0$	Intrinsic camera calibration parameters
u, v	Pixel image coordinates
$\mathbf{x} = [u,\, v,\, 1]^T$	Homogeneous coordinates of a point in the image plane
$\mathbf{x}_{(u)},\, \mathbf{x}_{(v)},\, \mathbf{x}_{(u,v)}$	The u-, the v- and the $u, v-$ pixel coordinates of the image point $\mathbf{x}$
$\mathbf{I}$	Photometrical matrix representation of an image
$\mathbf{I}(u, v)$	Pixel value of the image $\mathbf{I}$ at the coordinates (u, v)
$\mathbf{I}(u, v, t)$	Pixel value of the image $\mathbf{I}$ at time t located at (u, v)

$\mathbf{F}$	The fundamental matrix
$\boldsymbol{\nabla}$	The gradient operator
${}^{\mathcal{C}}X$, ${}^{\mathcal{C}}Y$, ${}^{\mathcal{C}}Z$	3D axis of the camera frame $\mathcal{C}$
$\mathbf{P}$	The projective camera matrix
$\mathbf{P}_0$	The canonical form of projective camera matrix
$\mathbf{D}_d$	The photometrical matrix representation of the disparity map
$\mathbf{H}_\Omega$, $\mathbf{H}_\infty$	A 2D- Homography matrix of a plane Ω and the infinite homography
$\mathbf{W}$	The weighting matrix
$\rho(\cdot)$	A weight function

Localization and Ego-Motion

$\mathcal{W}$, $\mathcal{E}(t)$	World- and Ego- (Body) reference frames
$\mathbf{x}(t)$	Vehicle ego-motion state vector
$\mathbf{x}(t\|t-1)$	Predicted or expected state
$\mathbf{P}(t)$	Covariance matrix of the vehicle ego-motion state
$\tilde{\mathbf{x}}(t)$	Observed stated or measure
$\mathbf{x} \sim \mathcal{N}\{\bar{\mathbf{x}}, \mathbf{P}\}$	The vehicle ego-motion, with its mean and covariance matrix
$\mathbf{A}$	Ego vehicle state transition model
$\mathbf{H}$	Ego vehicle observation matrix
${}^{\mathcal{W}}[\mathbf{q}(t), \mathbf{p}(t)]$	Attitude and position of the vehicle in $\mathcal{W}$

Moving Objects Tracking

$\mathbf{y}(t)$	Tracked object state vector (i.e. position and velocity)
$\mathbf{M}(t)$	Covariance matrix of the tracked object state
$\tilde{\mathbf{y}}(t)$	Object measurement (i.e. observation)
$\mathbf{N}$	Covariance matrix of the measurement noise
$\mathbf{A}_T$	State transition model
$\mathbf{C}_T$	Observation matrix
$\mathbf{O}$	Covariance matrix of the state transition error noise

Acronyms

ABS	Anti-lock Brake System
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
CAN	Controller Area Network
CCD	Charge Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
DARPA	Defense Advanced Research Projects Agency
DOF	Degrees Of Freedom
DoG	Difference of Gaussians
ECEF	Earth-Centered Earth-Fixed
ENU	East North Up
ESP	Electronic Stability Program
FA	False Alarms
FN	False Negative
FOV	Field Of View
FP	False Positive
GIS	Geographical Information System
GNN	Global Nearest Neighbor
GPS	Global Positioning System
IAC	Image of the Absolute Conic
ICP	Imaged Circular Points
ICP	Iterative Closest Point
IMM	Interacting Multiple Model
IMU	Inertial Measurement Units
IR	Infra Red
IRLS	Iteratively Re-weighted Least Squares
IV	Intelligent Vehicle
JPDA	Joint Probabilistic Data Association
JPDAF	Joint Probabilistic Data Association Filtering
LIDAR	LIght Detection And Ranging

LK	Lucas-Kanade
LM	Levenberg-Marquard
LTP	Local Tangent Plane
MHT	Multiple Hypothesis Tracking
ML	Multi-Layer
MSE	Mean Squared Error
MTT	Multiple Target Tracking
NCC	Normalized Cross-Correlation
NN	Nearest Neighbor
ORD	Ordering Constraint
PDA	Probabilistic Data Association
ROC	Receiver Operating Characteristic
ROI	Region Of Interest
ROU	Region Of Uncertainty
SAD	Sum of Absolute Differences
SBAS	Satellite Based Augmentation System
SIFT	Scale Invariant Features Transform
SSD	Sum of Squared Differences
STT	Single Target Tracking
SURF	Speeded-Up Robust Features
SVD	Singular Value Decomposition
SVS	Stereo Vision system
TN	True Negative
TOF	Time Of Flight
TP	True Positive
UV	Ultra Violet
WGS-84	World Geodetic System
WSS	Wheel Speed Sensors
ZNCC	Zero-mean Normalized Cross-Correlation
ZOI	Zone Of Interest

List of Figures

General Introduction

Right behind almost every research work, no matter the subject, there is always a social context which motivates it. Hereafter, this thesis gets into one of the actual subjects of our modern civilization: Intelligent vehicles and safety for driving assistance. In this huge context, we focus on traffic accidents problem in dynamic and complex urban environments. So far, vehicle accidental statistics confirm, every year, that human errors are the main cause of road accidents [SW09].

From above, two major research trends can be highlighted because of their contributions to address this important issue.

The first one is the development of autonomous cars. This advanced technology would not only potentially reduce traffic accidents, but also considerably improve the global energy consumption and the comfort of future passengers.

Researches on self-driving cars have revealed, in the past years, contributions in three widely known DARPA[1]-sponsored competitions: Two "Grand Challenges" and a "Urban Challenge" (see Fig. 1). These competitions have enlarged the scope and boosted the production of new researches in perception, control and motion planning. These areas cover probabilistic localization methods, mapping techniques, tracking strategies, global and local planning and the decision management through behavioral hierarchies. The DARPA challenges have demonstrated that embedded robotic systems can, nowadays, completely operate a passenger car traveling over significant distances and manage complex situations arising in quasi-urban real conditions.

The introduction of new sensory solutions was also an achievement of the last competition, proving the effectiveness of a high definition lidar in environment perception tasks. This sensor, namely Velodyne, was embedded in five of the six finalists cars, providing, in a 120m range, a full 360° horizontal field-of-view (FOV) and a 26.8° vertical FOV sampled in 64 layers. The cost, size and design of this sensor, however, situate this technology, for the moment, far away from a near feasible passenger car integration.

[1] *Defense Advanced Research Projects Agency*

(a) *Junior: Stanford Racing Team*

(b) *MIG: Freie Universität Berlin*

Figure 1: *Examples of autonomous vehicles*[2]

The second research trend concerns Advanced Driving Assistance Systems (ADAS). It stands for the deployment of active car safety systems that can efficiently assist users, while preserving their pleasure to drive. In contrast to autonomous vehicles, ADAS functionalities are mainly oriented towards proprio- and extero- perception understanding tasks which support and warn the driver by emitting alarms in presumed dangerous situations. Imminent harmful situations automatically trigger appropriate actions in the vehicle actuators (e.g. breaking) or passive systems (e.g. airbags, belt-tensioners, emergency call), in order to preserve the safety of the vehicle's occupants and of the involved road users.

Scene analysis and understanding performed by ADAS functionalities make use of multiple sensors. Naturally, the use and the interaction of multiple sensing principles is motivated by the complementarity of the perception means and the advantages of exploiting redundant information.

Nowadays, automotive manufacturers and suppliers commercialize innovative systems like Adaptive Cruise Control (ACC), forward collision warning, speed regulation, blind spot monitoring and lane departure warning. These functionalities are fundamentally based on the fusion of data provided by wheel speed sensors (WSS), gyroscopes and two innovative sensor technologies: Radars and cameras.

Since they achieve precise positioning and speed measurements of forward objects, radars (hyper frequency) are actually the most exploited sensors which are suitable for assistance tasks in high dynamical environments (e.g. highways) such as ACC and collision warning.

Cameras are passive sensors which are more and more used in ADAS, like blind spot monitoring, lane departure warning, speed limit signals and pedestrian recognition. The large functional spectrum offered by this sensor makes it quite attractive, since it can replace existing on-board sensors or provide redundancy in safety applications (see Fig. 2).

[2]Images from [SKG09] and *www.autonomos-labs.de* repectively

(a) *Volvo City Safety Application based on Vision* **(b)** *Continental Lane Departure Warning Assistance* **(c)** *Mobileye: Forward collision warning*

Figure 2: *Examples of Advanced Driver Assistance Systems* [3]

In academical research, lasers (e.g. lidar) constitute a promising sensor technology which grants access to accurate measurements of the scene geometry, reliable in poor weather conditions. This information is crucial for object detection and tracking applications in dense traffic conditions. Alternatively, multiple camera systems are able to provide full 3D range measurements at a high rate, registered with dense photometrical information. This visual scene geometry information is rarely exploited in ADAS. In this context, the interaction between lidar and cameras seems to be favorable for the improvement of existing lidar-based multiple target tracking applications.

Accordingly, our research addresses the study of multiple cameras capabilities (here stereo vision) by exploring their potential contribution to the scene geometry perception in a multi-sensor object localization and tracking scheme for Intelligent Vehicles. A possible solution to this issue could be to detect and to track objects based on lidar measurements only. As lidar measurements are referenced to a mobile platform, the dynamics of an observed object and its trajectory are complex and rarely suitable to be linearly approximated. Wheel speed sensors can be used to compensate the vehicle's motion, achieving a better localization and tracking of the observed objects with respect to a fixed reference frame.

In order to enable geometrical data exchanges between a lidar and a vision system, we firstly deploy, in this thesis project, a new calibration technique. This link is established through the estimation of the extrinsic parameters and their corresponding confidence intervals.

In order to determine the real ego-trajectory in space, we also estimate, in real-time, the accurate 3D motion of the vehicle relying only on stereo images. The proposed method is then considerably improved through the fusion of wheel speed sensors and yaw-rate gyrometer measurements. This positioning solution can constitute an alternative to

[3]Images from *www.volvo.com, www.conti-online.com and www.mobileye.com*

the widely spread navigation systems based on GPS[4] receivers.

Considering that the trust accorded by the driver to an ADAS is extremely important, we propose a self-assessment function to verify the existence of tracked objects through visual confirmation. In this way, the integrity of the perceived environment is increased as false alarms are reduced.

Since the asynchronicity of multi-perception functions in real life conditions is a problem that has a great impact on the fusion, a synchronization strategy is also proposed based on the use of predictive filters. This solution preserves the temporal integrity of the fusion data process.

Finally, the complete multi-modal object localization and tracking system is deployed and evaluated in real world conditions, using stereo vision to estimate the vehicle's motion and to simultaneously confirm the existence of tracked objects. This system is able to provide a dynamic map containing the kinematic states of the surrounding objects in the scene.

Outline of the manuscript

This thesis contains a detailed description of the methods, theoretical concepts, experimental observations, results and conclusions of this research. Every chapter constitutes a small step towards the implementation of a complete multi-modal object localization and tracking system enhanced by vision. Three major chapters enclose all this work: Multi-Modal Perception System, Vision-based Odometry and Multi-Modal Object Localization and Tracking.

Multi-Modal Perception System

As our research is directly related to the considered experimental platform, this chapter starts by providing a detailed description of the involved perception means. Additionally, a calibration methodology between the lidar and the vision system is proposed. The obtained results are used as a link, allowing us to merge the geometrical perception of both sensors in a common reference frame. These results are consequently used in the further experiments of this research.

Vision-based Odometry

The motion estimation from stereo image sequences requires the knowledge of the multiple view geometry induced by the image formation process. These theoretical principles are presented as well as the formal development of the 3D motion estimation

[4]Global Positioning Systems

loop. Based on the incremental integration of elementary displacements, this chapter presents an alternative localization method able to provide the position and the attitude of the vehicle in the 3D space. The last part of this chapter is devoted to the presentation of a strategy to provide a GPS-like positioning based on the proposed odometry method.

Multi-Modal Object Localization and Tracking

Herein, the complete multiple target tracking system is detailed. The geometrical scene information provided by the vision system is integrated in a self-assessment function which confirms the existence of the tracked objects. This function aims at increasing the integrity of the track hypothesis and to reduce the false alarms issued by the system. This chapter also addresses the multi-rate information management required when dealing with asynchronous perception functions. The management of the system asynchronism ensures the temporal consistency of the fused data.

The document is concluded by a synthesis of the addressed problems, the contributions and some research perspectives.

Chapter 1

Multi-Modal Perception System

Contents

1.1 Introduction

Perception is a complex task in variable and non structured environments, even for human beings. Naturally, humans can fuse their five senses to infer and solve complex reasoning tasks. The concept of fusing multiple perception sources to achieve a more specific inferences has been applied for several years, mainly in robotic applications. This knowledge and advancements are now applied to "assist" drivers on-board. Since an Intelligent Vehicle (IV) has to perceive, to monitor and to understand every single change in its own state and in the surroundings, multiple sensory principles have to be exploited.

Combining multiple information sources does not only have a complementarity objec-

tive. Redundancy is also a statistical advantage obtained, for instance, through the combination of sensors having the same field-of-view.

In this context, this chapter presents, in section 1.2, a detailed description of the experimental platform considered in this study, its perception capabilities and sensor technology. Additionally, in-depth details about the geometrical sensor models and calibration procedure used specifically for lidar and vision sensors are also provided in section 1.3. The calibration proposed technique is intended to ensure the consistency and the quality of the merged information and to quantify the uncertainty intervals of estimated results.

1.2 Sensing capabilities of Carmen

An experimental vehicle belonging to the Heudiasyc Laboratory is used for the implementation and the validation of the global perception functions in real-life conditions. The platform is called Carmen and is equipped with a long-range radar, a Velodyne lidar scanner, GPS receivers, a stereo and a monocular vision system, a multi-layer lidar and a CAN-bus[1] gateway which grants access to embedded sensors used by built-in ABS[2] and ESP[3] functions: Wheel Speed Sensors (WSS) and yaw rate gyroscope.

For computation, Carmen uses a central unit equipped of a single 3.1 GHz Core Duo Quad processor with 4GB of memory and a RAID[4]0 of four hard disks providing 2TB for data logging.

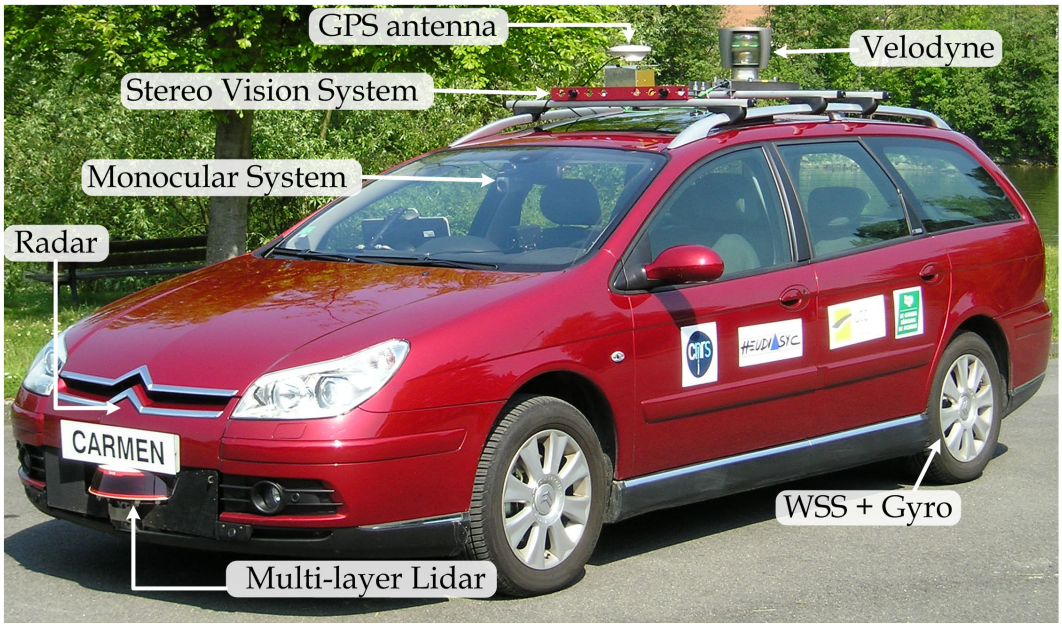
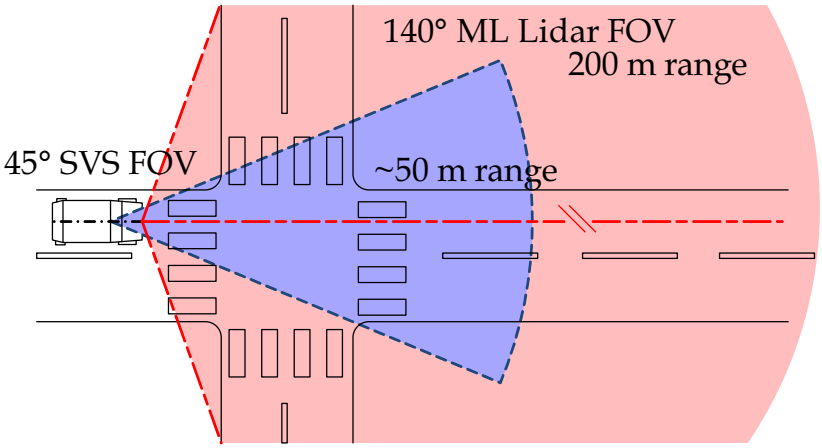


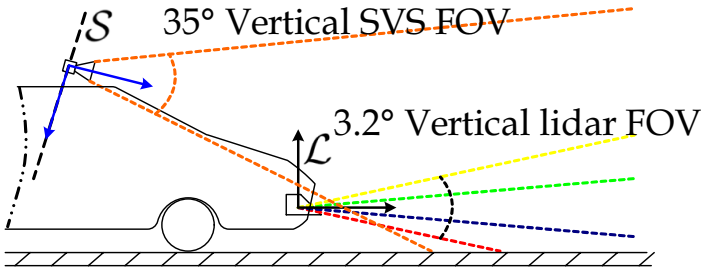
Figure 1.1: *Test-bed vehicle*

[1]Controller Area Network
[2]Anti-lock Braking System
[3]Electronic Stability Program
[4]Redundant Array of Independent Disks

The multi-sensor system considered in this study is composed of a multi-layer lidar, cameras rigidly fixed to the vehicle (i.e. monocular and stereo vision systems) and the WSS-Gyro measurements. The ML lidar is installed at the front of the vehicle in the bumper section and the cameras are located behind the windshield and on the top of the vehicle (see Fig. 1.1). This sensor configuration only presents some occlusions to the camera for short distances with respect to the ML-lidar. Layout representations of the field-of-view coverage provided by the exteroceptive sensors are illustrated in Fig. 1.2.



(a) *Layout representation of the horizontal perception coverage provided by the stereo vision system (SVS) and the ML lidar*



(b) *Layout representation of the vertical perception field-of-view. The multi-layer divergence of the lidar is exaggerated for clarity.*

Figure 1.2: *Field-of-view coverage of the experimental vehicle*

In the following, detailed information of the considered sensors is presented, addressing the perception principle, the sensor specifications and their typical automotive applications.

1.2.1 Multi-layer Lidar

Laser range sensors constitute an interesting active mean for measuring the relative distance of a remote target. There are three major sensing principles for this measurement instrument: Triangulation, phase modulation and time-of-flight (TOF).

Automotive and robotic range finders are usually TOF-based sensors which have been widely exploited, principally, to accomplish many perception tasks, like mapping and obstacle detection. The estimation of range distances, in this case, is achieved by measuring the elapsed flight time of a pulsed light to the target and back. The distance is then computed as $l = (t\,c)/2$, where t is the measured TOF and c the speed of light in the given medium. Considering that the sensor emits a pulsed light, the measurement uncertainty is given by $\Delta l = (c\,\Delta t)/2$ where Δt is the pulse width.

As illustrated in Fig. 1.3, laser range sensors often use a horizontally rotating mirror, which deflects the laser beam in different angles. This configuration achieves a range scan measurement of the scene surroundings.

A common advantage of laser range sensors is that they can provide not only range but also reflectance information. This latter grants meaningful knowledge about the object surface appearance.

Concerning their drawbacks, two major problems can be listed. First, range sensors are based on a diffuse light beam reflection assumption. Thus, in case of specular surfaces (e.g. mirrors, polished metal, water, and so on), the sensor receiver can detect light beam reflections, which have potentially followed multiple paths. These measurements entail incorrect or missing range echos. Second, the laser beam spot is subject to diffraction effects, so it can partially impact two surfaces placed at different distances. In such a case, the sensor usually reports a spurious range corresponding to the mean distance of both detected objects.

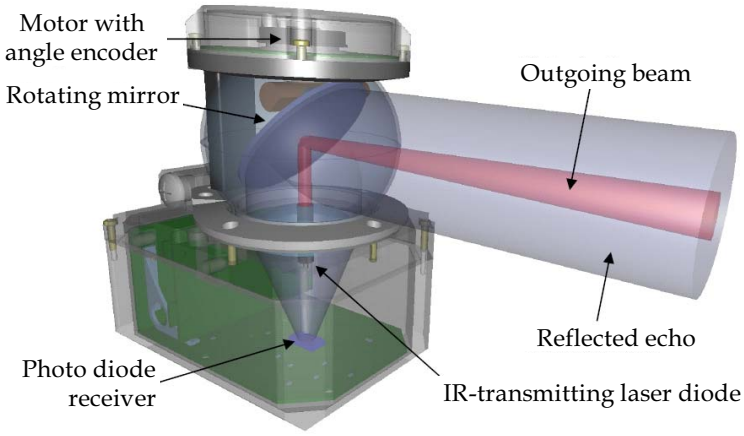


Figure 1.3: *Laser range finder internals*[5]

LIDAR technology

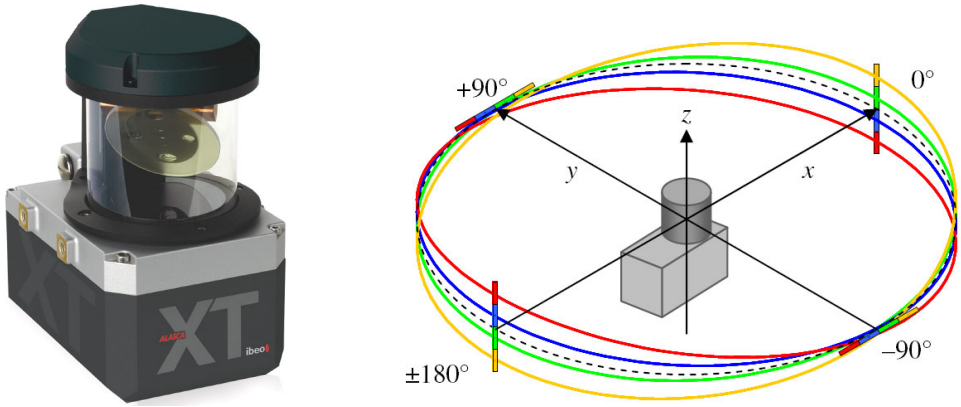
LIght Detection And Ranging (LIDAR) is a technology similar to the one presented for telemeters. In contrast, LIDAR sensors are characterized by the emission of a pulsed

[5]Image taken from IBEO Alasca XT user manual

light source with high peak power and a wavelength range which spans from near-UV (300 nm) to far IR (10 μm). In addition, the delay and attenuation of the received echos can be exploited to determine, for instance, if the laser reflection comes from a water drop, an object or the ground.

Specifications of the lidar sensor

Nowadays, one of the most adapted lidar design for automotive applications is the Ibeo Alasca XT. Equipped with a rotating mirror and four independent photodiode receivers, this sensor is able to report 200m range measurements from four different detection planes also called layers (see Fig. 1.4).



(a) *Ibeo Alasca XT*

(b) *The Ibeo Alasca XT measures ranges in four detection planes (i.e. layer) with different directions*

Figure 1.4: *Ibeo Alasca XT* [6]

These four crossed scan layers cover a vertical field-of-view of 3.2° where the inter-layer divergence corresponds to 0.8°. The advertised light beam divergence is 0.5 mrad which is equivalent to a 50 cm diameter spot at a 100m range. The multi-layer technology is intended to ensure object detection in case of vehicle pitch changes (assuming that the sensor is mounted at the front of the vehicle).

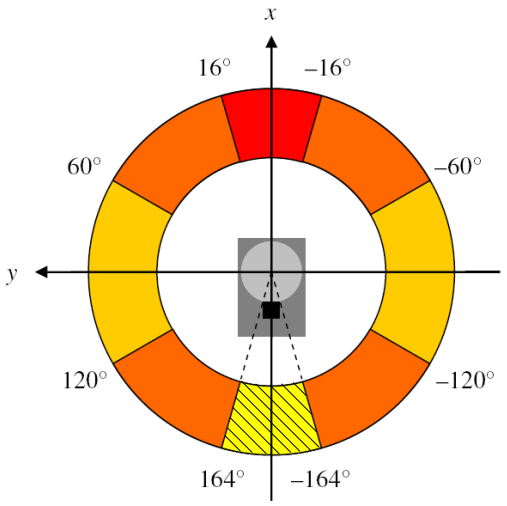
This TOF sensor is also equipped with a multi-target technology allowing the reception of up to four echoes per transmitted pulse. In this way, the lidar sensor can eliminate, sometimes, potential artifact reflections arising from a dirty cover or poor weather conditions (e.g. fog or heavy rain).

The supported scanning frequencies vary from 8 to 40 Hz, but optimized operation is achieved for rotation frequencies of 12.5 Hz and 25 Hz. The angular resolution of the sensor is directly related to the scanning period, it can thus be enhanced by increasing the scan duration, within an eye-safe operating limit. For the specific case of 12.5 and

[6] Images taken from Ibeo Alasca XT user manual document

25 Hz scan frequencies, the angular resolution is illustrated in Fig. 1.5, covering a total horizontal FOV of almost 330°.

Typically, the multi-layer lidar is mounted either at the front (center or left/right corners) or the rear of the vehicle for applications such as pedestrian and obstacle detection, lane departure warning and ACC (acronym of *Adaptive Cruise Control*) functionalities and autonomous navigation tasks [SF06]. Recently in [MON09], the ML-lidar perception capabilities were exploited for a simultaneous ego-motion estimation and object detection and tracking.



Resolution	
12.5 Hz	25 Hz
0.125°	0.25°
0.25°	0.5°
0.5°	1°
1°	2°

Figure 1.5: *Ibeo Alasca XT field-of-view and angular resolution* [7]

1.2.2 Vision system

A vision system is a passive exteroceptive perception mean which grants access to a huge quantity of photometrical and geometrical information represented on an image. The existence of this great information source is one of the reasons why the machine vision field is a vast discipline where different approaches of widespread applications take advantage of this mean; Robotics is not an exception. As vision systems provides a large functional spectrum at a relative low cost, car manufactures and suppliers of the automotive industry have recently focused their attention on it for the development of Advance Driver Assistance Systems (ADAS).

Sensor imager technology

An image is represented by a bi-dimensional array where each element, called *pixel*, takes positive values over a discrete brightness (i.e. irradiance) range (typically 0 to 255). This representation is an abstraction of a rectangular planar photosensitive

[7] Image taken from Ibeo Alasca XT user manual information document

sensor, known as the imager. In a gray-scaled image each single photo detector maps into a pixel value. For the particular case of color imagers, four interlaced single photo detectors reconstitute a pixel, composed by the three basic components of the visible spectrum (i.e. red-green-blue). This is done through the use of a color filter, referred to as the Bayer filter.

The most known sensor technology is the charge-coupled device (CCD) which is a passive-pixel sensor. The most energy-efficient and cheapest technology, however, is the CMOS (Complementary Metal-Oxide-Semiconductor) image sensor, in which photo detectors are coupled with an active amplifier. Contrarily to the CCD, the latter technology allows the integration of image processing functions which are built-in the imager chip.

Since images are formed by the total exposition of the photosensitive sensor during a determined integration time, vision systems are drastically sensible to changing illumination conditions induced principally by the weather or the dynamic content of the observed scene. To mitigate this drawback, a controlled acquisition method called rolling shutter, consists in exposing partially and progressively the pixel-sensor, scanning the whole image. Like the laser scanner, this technique can induce spatial and temporal image aliasing in extreme dynamic conditions.

Imaging sensors

The concepts presented so far, are applicable for a large variety of imaging sensors, commonly called cameras. The most traditional one is the perspective camera which is composed of an optical system, focusing the light right on the imager. The study of the image formation process in the perspective cameras is well known and describes how observed objects give rise to images. Moreover, computer vision addresses the inverse problem so as to recover a description of objects in space from images. Such an object description could be obtained either semantically, like object recognition based on photometrical information, or geometrically based on the scene structure [BSFC08].

A perspective camera covers typically a 45° horizontal field-of-view. Some automotive applications, however, require to cover larger blind-zones [HGJD09]. To this end, a perspective camera can be modified by the use of a wide-lens, creating a so-called fish-eye system (i.e. dioptric camera). Fish-eye cameras can cover up to 180° horizontal field-of-view, but radial lens distortions cause a nonlinear pixel mapping of the image plane. This complexifies image processes in applications like object detection, recognition and classification.

Another interesting imaging sensor is the catadioptric camera (also called omnidirectional camera). This variant is obtained with a perspective camera and a convex mirror and its major advantage is the coverage of a hemisphere field-of-view in a single image. Mobile navigation applications exploit quite well the appropriate information provided

by this perception mean [SFS09, Mei07]. It is worth mentioning that images obtained with the use of catadioptric cameras are characterized by a low resolution and a central blind spot. A sample image of each presented imaging sensor is shown in Fig. 1.6.

Figure 1.6: *Perspective, dioptric*[8] *and catadioptric*[9] *images*

The combination of two projective cameras with overlapping field-of-views engenders a stereo imaging system able to provide full 3D range measurements, registered with dense photometrical information at a high rate (up to 30 fps). This can not be achieved by other existing range sensor.

Based on the ideal geometrical configuration illustrated in Fig. 1.7, stereo imaging systems can *triangulate* the 3D localization of a point knowing the baseline distance between the cameras, the projective cameras' parameters (i.e. focal distance and image center point) and the location of the observed point in both images. This last requirement constitutes the main difficulty to ensure 3D reliable estimates as those provided by a lidar range sensor. In-depth details about the projective camera model and the two-view geometry will be provided in further sections.

Figure 1.7: *Stereo vision perception* [10]

[8]Dioptric image taken from [HGJD09]
[9]Catadioptric image taken from [GT05]

Automotive experiences reported by Dickmanns [Dic07] in the use of stereo vision over 5000 km of fully autonomous driving in normal traffic on German and French freeways show the viability of this kind of system. Typically installed looking forward close to the rear-view mirror, vision systems have shown a great potential in applications such as distance keeping (platooning), lane departure warning, intelligent headlamp control, emergency brake assistance, obstacle detection and pedestrian and traffic sign recognition.

Specifications of the monocular and stereo vision system

The considered monocular system is composed of a perspective camera Sony DFW-VL500 (see Fig. 1.8a). This imaging sensor is configured to deliver 640×480 pixel gray-scaled images acquired with a CCD imager and an integral 12x motorized zoom lens. This camera is used particularly in camera-lidar calibration tests or just as a visual reference.

The installed stereo vision system (SVS) is a Videre STH-MDCS-VARX (see Fig. 1.8b) which provides a wide variable baseline rig set up to 47cm. This system is composed of two CMOS cameras with 4.5mm lenses and is configured to acquire 320×240 pixels gray-scale images at 30 fps. Equipped with synchronized imagers, this SVS ensures the exposure of corresponding pixels at exactly the same time.

(a) *Perspective CCD camera Sony DFW-VL500*

(b) *Videre STH-MDCS-VARX composed of CMOS perspective cameras*

Figure 1.8: *Monocular and stereo vision systems on-board Carmen*

1.3 Geometrical calibration

As said previously, each range sensor (Stereo vision and Lidar) used in our perception system will provide range information (3D depth points) in its own local frame. Therefore, if one wants to fuse the geometrical information supplied by each sensor

[10] Original image source from database of the MOBILE3DTV project funded by the Europe's Seventh Framework Programme

in a cooperative way, one has to estimate the relative pose (misalignment or calibration) between these two sensors. Different calibration methods have been proposed to estimate with accuracy, the rigid transformation (6 degrees of freedom corresponding to a rotation matrix and a translation vector) between a camera and a range sensor. Zhang et al. [ZP04] have proposed a target-based calibration method between a camera and a single-row laser range finder. This method was considerably improved by Dupont et al. in [DKF05]. A target-less calibration approach for a 3D laser range finder has been presented by Scaramuzza et al. in [SHS07]. In the same way, Zhao et al. in [ZXJ+09] define the calibration procedure as a sensor alignment problem using geometric primitives defined in both perception modalities. Then, the user specifies which primitives to associate and the process estimates the relative motion between the sensors. In our case, we focus on the use of a calibration target, since we want to obtain an automatic calibration procedure with minimal user interaction. The choice of a suitable multi-modal calibration object is complicated: It has to be easy to detect by each sensor and its geometrical properties must provide 3-D relevant information so as to obtain a full 3-D extrinsic calibration. Base on these criteria, we propose a method which takes advantage of visual information and ML lidar data (i.e. four layers) by using a circle-based calibration target. Indeed, this kind of target is easy to detect and projective properties of conics can be used to estimate both internal camera and pose parameters. The circle-based calibration target is a rigid plane with a printed black ring where the inner circle describes a plane perforation (see Fig. 1.9).

1.3.1 Principle of the method

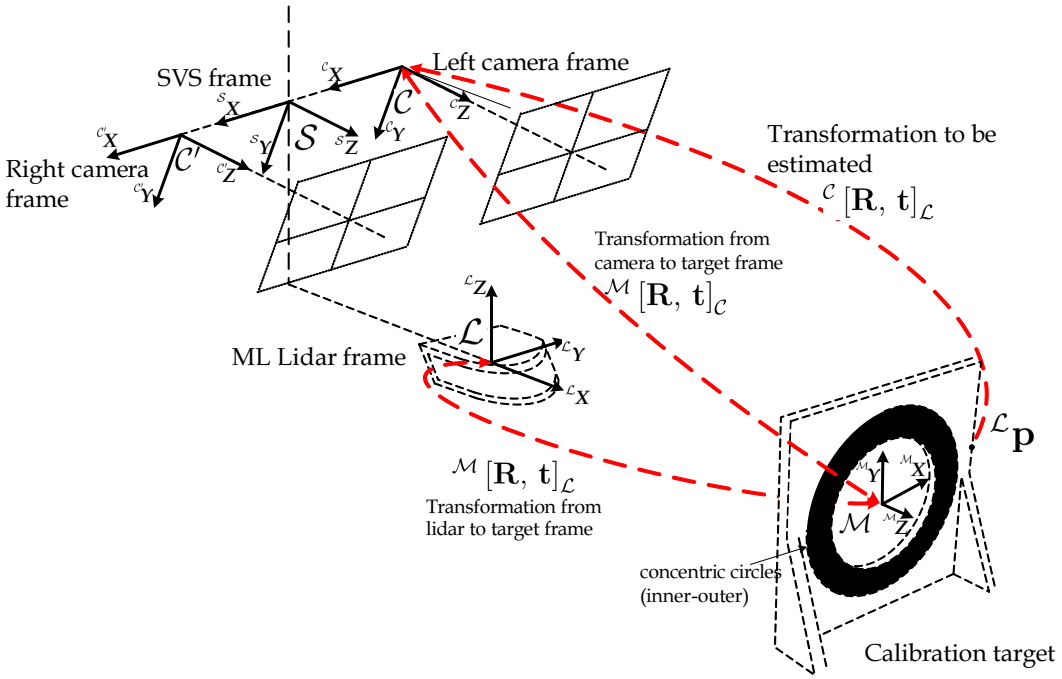


Figure 1.9: *Frames involved in the lidar-camera calibration*

The purpose of the proposed approach is to estimate the rigid body transformation, denoted ${}^{\mathcal{C}}[\mathbf{R}, \mathbf{t}]_{\mathcal{L}}$, between the ML lidar and one of the SVS frames (here, the left camera frame) using a shape-constrained calibration target (See Fig. 1.9). Hence, the composition of the partial transformations between ML lidar - target and left camera - target corresponds to the direct mapping between the Euclidean spaces of the ML lidar and the left camera. For this, let $\mathcal{M}$ be the circular calibration target frame, $\mathcal{C}$ ($\mathcal{C}'$) the left (right) camera frame and $\mathcal{L}$ the lidar frame. Thus, a 3D point $\mathbf{p}$ can be defined in each frame by ${}^{\mathcal{X}}\mathbf{p}$ where $\mathcal{X}$ can be one of the frames $\mathcal{M}$, $\mathcal{L}$, $\mathcal{S}$, $\mathcal{C}$ or $\mathcal{C}'$.

Hereafter, the proposed method is detailed in four major steps which cover the target pose estimation in the lidar and in the camera frame, the computation of the calibration parameters and the estimation of accuracy indicators.

1.3.2 Target pose estimation in the lidar sensor frame

This section briefly presents the model adopted for the geometrical perception granted by the ML lidar. The identification and the localization of the target pose based on the lidar measurements is then detailed.

Geometrical model of the Lidar

As presented in section 1.2.1, a multi-layer lidar sensor provides 4 crossed-scan-planes. Each scan-plane has a relative altitude aperture and an azimuthal angle resolution. A 3D laser impact can be defined in the lidar frame $\mathcal{L}$, as: ${}^{\mathcal{L}}\mathbf{p} = \left[{}^{\mathcal{L}}\mathbf{p}_{(x)}, {}^{\mathcal{L}}\mathbf{p}_{(y)}, {}^{\mathcal{L}}\mathbf{p}_{(z)}\right]^T$. In order to express the location any lidar point with respect to the calibration object frame (i.e. target frame, $\mathcal{M}$), it is necessary to define a rigid geometric transformation composed of a rotation and a translation ${}^{\mathcal{M}}[\mathbf{R}, \mathbf{t}]_{\mathcal{L}}$. Therefore, the corresponding point of ${}^{\mathcal{L}}\mathbf{p}$ in the target frame $\mathcal{M}$, noted ${}^{\mathcal{M}}\mathbf{p}$, can be obtained by applying the following rigid transformation:

$$ {}^{\mathcal{M}}\mathbf{p} = {}^{\mathcal{M}}\mathbf{R}_{\mathcal{L}}.{}^{\mathcal{L}}\mathbf{p} + {}^{\mathcal{M}}\mathbf{t}_{\mathcal{L}} \tag{1.1} $$

The scan of lidar beams describes the inner perforation border as a 3D circle shape. A robust circle detection is achieved using of several lidar scans of the target. Then, outlier data can be filtered using the technique proposed in [DKF05] and applied on n 4-layer lidar scans of the calibration scene, generating 8 points per pose. Using the filtered data, points lying in the perforation border of the calibration target are extracted using a 1-D edge detection. The detected border points are used in a circle fitting algorithm [GGS94] obtaining in this way the normal ${}^{\mathcal{L}}\mathbf{n}$ to the supporting plane of the circle and its center ${}^{\mathcal{L}}\mathbf{c}$.

Using Eq. 1.1, it is possible to obtain the pose of target frame origin with respect to the lidar frame by ${}^{\mathcal{L}}\mathbf{c} = -({}^{\mathcal{M}}\mathbf{R}_{\mathcal{L}}^T.{}^{\mathcal{M}}\mathbf{t}_{\mathcal{L}})$. Taking advantage of this fact, one can perform a

nonlinear 3D circle fitting problem constrained to a known radius, r, and parametrized as follows: $\hat{\alpha}$, $\hat{\beta}$ are the orientation angles of the 3D circle axis vector, ${}^{\mathcal{L}}\mathbf{n}_{\left(\hat{\alpha},\hat{\beta}\right)}$, with respect to y-axis and z-axis respectively (See Fig. 1.9) and ${}^{\mathcal{L}}\hat{\mathbf{c}} = \left[{}^{\mathcal{L}}\hat{\mathbf{c}}_{(x)},\, {}^{\mathcal{L}}\hat{\mathbf{c}}_{(y)},\, {}^{\mathcal{L}}\hat{\mathbf{c}}_{(z)}\right]^T$ are the Cartesian coordinates of the estimated 3D circle center coincident with the target frame origin. As proposed in [Sha98], it is possible to estimate the normal vector ${}^{\mathcal{L}}\mathbf{n}_{\left(\hat{\alpha},\hat{\beta}\right)}$ and the circle center ${}^{\mathcal{L}}\hat{\mathbf{c}}$ using a non linear minimization over the following geometrical criterion:

$$e = \sum_{i=1}^{8n}[A_i^2 + B_i^2] \tag{1.2}$$

with

$$A_i = {}^{\mathcal{L}}\mathbf{n}_{\left(\hat{\alpha},\hat{\beta}\right)} \cdot \left({}^{\mathcal{L}}\mathbf{p}_i - {}^{\mathcal{L}}\hat{\mathbf{c}}\right) \tag{1.3}$$

$$B_i = \| {}^{\mathcal{L}}\mathbf{n}_{\left(\hat{\alpha},\hat{\beta}\right)} \times \left({}^{\mathcal{L}}\mathbf{p}_i - {}^{\mathcal{L}}\hat{\mathbf{c}}\right) \| - r \tag{1.4}$$

where the dot and cross product operators are denoted respectively $(\cdot)$ and $(\times)$, and:

- A_i corresponds to the Euclidean distance between a target-contour ML lidar impact, ${}^{\mathcal{L}}\mathbf{p}_i$, and the 3D plane defined by ${}^{\mathcal{L}}\mathbf{n}_{\left(\hat{\alpha},\hat{\beta}\right)}$ and the estimated circle center, ${}^{\mathcal{L}}\hat{\mathbf{c}}$.
- B_i represents the Euclidean distance between a target-contour ML lidar point, ${}^{\mathcal{L}}\mathbf{p}_i$, and the 3D circle axis defined by ${}^{\mathcal{L}}\mathbf{n}_{\left(\hat{\alpha},\hat{\beta}\right)}$ and the estimated circle center, ${}^{\mathcal{L}}\hat{\mathbf{c}}$.

Accordingly, the criterion in Eq. 1.2 is minimized making use of the Levenberg-Marquardt algorithm (LM-algorithm) [Mar63]. After convergence of the non linear minimization algorithm and by applying this technique to various poses of the calibration target, we obtain not only a first set of 3D laser features (i.e. circle centers, ${}^{\mathcal{L}}\hat{\mathbf{c}}$, and normal plane vectors, ${}^{\mathcal{L}}\mathbf{n}$) but also a 3D circle reconstruction in the ML lidar frame for every pose.

1.3.3 Target pose estimation in the vision sensor frame

In contrast to the lidar case, a camera is a passive sensor. As stated before, its perception principle relies on the convergence of the light onto its imager. This study consider the image formation process through an ideal pinhole camera model. In the sequel the ideal perspective projection concepts are provided and a strategy to estimate the target pose in space by the means of two imaged concentric circles is proposed.

Model of the vision Sensor

Let ${}^{\mathcal{M}}\mathbf{p} = [X,\, Y,\, Z]^T$ be a 3D point in the target frame $\mathcal{M}$ and $\mathbf{x} = [u,\, v,\, 1]^T$ the 2D homogeneous coordinates of its projection in the image of the left camera frame $\mathcal{C}$ (see

Fig. 1.10). The relationship between ${}^{\mathcal{M}}\mathbf{p}$ and $\mathbf{x}$ is expressed as a perspective projection equation [HZ03],

$$\mathbf{x} \sim \mathbf{K}\left({}^{\mathcal{C}}\mathbf{R}_{\mathcal{M}} \,.^{\mathcal{M}}\, \mathbf{p} +^{\mathcal{C}}\, \mathbf{t}_{\mathcal{M}}\right) \tag{1.5}$$

We use the notation $\sim$ to denote an equality up to a scale factor. The matrix $\mathbf{K}$ is an upper triangular matrix called the intrinsic matrix [HZ03]:

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.6}$$

The parameters f_x, f_y, u_0 and v_0 are known as the intrinsic parameters (assuming zero skew) and represent respectively the focal lengths in the x and y directions, and the coordinates of the principal point. ${}^{\mathcal{C}}\mathbf{R}_{\mathcal{M}}$ and ${}^{\mathcal{C}}\mathbf{t}_{\mathcal{M}}$ are respectively the rotation matrix and the translation vector between the frames $\mathcal{M}$ and $\mathcal{C}$.

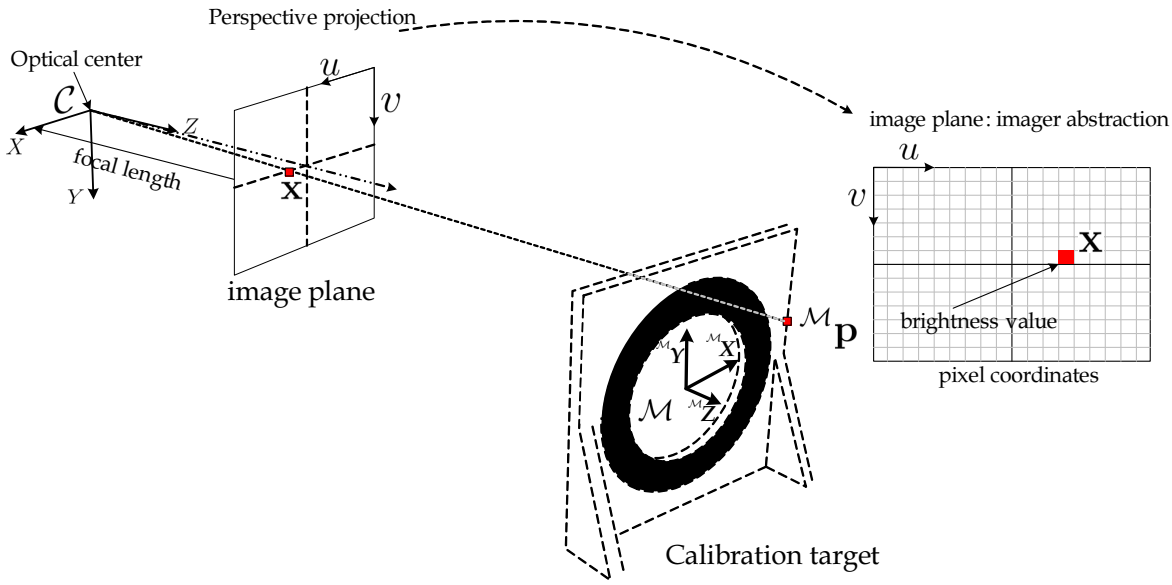


Figure 1.10: *Frontal pinhole model of image formation*

It is important to notice that prior to the intrinsic calibration, the lens distortions (especially radial distortion) have to be removed. Let (u, v) be the ideal (distortion-free) pixel image coordinates, which are obtained using Eq. 1.5, and $(\breve{u}, \breve{v})$ the corresponding real observed image coordinates.

$$\begin{aligned} \breve{u} &= u + (u - u_0)\,(k_1.r^2 + k_2.r^4) \\ \breve{v} &= v + (v - v_0)\,(k_1.r^2 + k_2.r^4) \end{aligned} \tag{1.7}$$

where k_1 and k_2 are the coefficients of the radial distortion, and $r^2 = \left(\frac{u-u_0}{f_x}\right)^2 + \left(\frac{v-v_0}{f_y}\right)^2$
The proposed circle-based calibration target is composed by two concentric circles.

Projective properties of conics make them suitable for camera calibration [KGK05, FC02, KO04]. It is therefore possible to infer metric properties about the camera (intrinsic parameters) and the scene (3D calibration target position) from the perspective projections of circular features.

Let define a reference plane in frame $\mathcal{M}$ with equation ${}^{\mathcal{M}}Z = 0$. The equation 1.5 becomes

$$\mathbf{x} \sim \mathbf{K}\,[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}]\,.{}^{\mathcal{M}}\underline{\mathbf{p}} \sim \mathbf{H}.{}^{\mathcal{M}}\underline{\mathbf{p}} \tag{1.8}$$

where ${}^{\mathcal{M}}\underline{\mathbf{p}} = [X,\ Y,\ 1]^T$ and $\mathbf{r}_1$, $\mathbf{r}_2$ are the first two columns of $\mathbf{R}$. The equation of a 3D circle of radius r centered at the origin of the frame $\mathcal{M}$ can be defined as:

$${}^{\mathcal{M}}\underline{\mathbf{p}}^T.\mathbf{Q}.{}^{\mathcal{M}}\underline{\mathbf{p}} = 0, \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -r^2 \end{bmatrix} \tag{1.9}$$

Using Eq. 1.8, the perspective projection of the circle $\mathbf{Q}$ gives a conic $\mathbf{C}$ which is defined by

$$\mathbf{x}^T\mathbf{C}\mathbf{x} = 0, \quad \mathbf{C} \sim \mathbf{H}^{-T}\mathbf{Q}\mathbf{H}^{-1} \tag{1.10}$$

As stated in [Zha00], the Imaged Circular Points (ICPs), which are the projections of the Circular Points (common to all circles including the projective invariant known as the Absolute Conic) are in the form of $\mathbf{e}_\pm = \mathbf{h_1} \pm i\mathbf{h_2}$ where $\mathbf{h_1}$ and $\mathbf{h_2}$ are the first two columns of the world-to-image homography $\mathbf{H}$.

Thanks to [Tri97], the Image of the Absolute Conic (IAC), denoted ω, encodes the intrinsic properties of the camera and is defined by

$$\omega \sim \mathbf{K}^{-T}\mathbf{K}^{-1} = \begin{bmatrix} \frac{1}{f_x^2} & 0 & \frac{-u_0}{f_x^2} \\ 0 & \frac{1}{f_y^2} & \frac{-v_0}{f_y^2} \\ \frac{-u_0}{f_x^2} & \frac{-v_0}{f_y^2} & \frac{u_0^2}{f_x^2} + \frac{v_0^2}{f_y^2} + 1 \end{bmatrix} \tag{1.11}$$

As stated in [KGK05], all intrinsic calibration parameters can be obtained first by computing the image of the absolute conic (IAC) with precision from the imaged circular points using at least three images of two concentric circles under different orientations. By using the method exposed in [KGK05] the intrinsic camera parameters denoted in 1.6 can be estimated.

If each ICP lies on the IAC, the following constraint is enforced:

$${\mathbf{e}_\pm}^T\omega\mathbf{e}_\mp \sim 0 \tag{1.12}$$

which can be rewritten as

$$\begin{aligned} \mathbf{h_1}^T \omega \mathbf{h_2} &= 0 \\ \mathbf{h_1}^T \omega \mathbf{h_1} - \mathbf{h_2}^T \omega \mathbf{h_2} &= 0 \end{aligned} \tag{1.13}$$

Each view of the target gives two equations which are linear in the upper diagonal elements of the IAC [HZ03]. Providing that $\mathbf{K}$ is constant, three views are necessary to obtain a solution. As presented in [KGK05], one ICP (and therefore its complex conjugate) can be recovered from one view of a pair of projected concentric circles $\{\mathbf{C_1}, \mathbf{C_2}\}$ computing the rank-2 matrix $\mathbf{\Delta_2}$:

$$\mathbf{\Delta_2} = \beta_2 \mathbf{C_1^{-1}} - \mathbf{C_2^{-1}} \sim \mathbf{H}\,\mathrm{diag}(0,\, 0,\, 1)\,\mathbf{H}^T \tag{1.14}$$

Then the ICPs are recovered in the form of $\sqrt{s_1}\mathbf{u_1} \pm i\sqrt{s_2}\mathbf{u_2}$, where $\mathbf{u_1}$ and $\mathbf{u_2}$ are the first two columns of $\mathbf{U}$ resulting from the SVD (Single Value Decomposition) of $\Delta_2 = \mathbf{U}\mathrm{diag}(s_1, s_2, 0)\mathbf{U}^T$ with $\mathbf{U}\mathbf{U}^T = \mathbf{I}$. The determination of $\mathbf{K}$, is performed by estimating the IAC, ω, as the locus of all ICPs, through a linear formulation of the problem (Eq. 1.13). Finally, a Cholesky factorization of the IAC gives the matrix $\mathbf{K}$. Thus far, all intrinsic parameters in Eq. 1.6 have been estimated. The next stage is now to obtain the relative transformation between the target and the camera frames.

As stated earlier, the considered target consists in two concentric circles $\mathbf{Q}_1$ (of radius r_1) and $\mathbf{Q}_2$ (of radius r_2). Those circles are projected on the image as conic curves $\mathbf{C_1}$ (exterior) and $\mathbf{C_2}$ (interior) (see Eq. 1.10). Both circles can be detected in the image using segmentation methods widely explained in [KO04, XJ02].

Now, let be $\mathbf{C}$, a generic conic obtained by the perspective projection of a 3D circle,

$$\mathbf{C} = \begin{bmatrix} A(f_x/f_y) & B/2 & D/2f_y \\ B/2 & C(f_x/f_y) & E/2f_x \\ D/2f_y & E/2f_x & F/(f_x f_y) \end{bmatrix} \tag{1.15}$$

where A, B, C, D, E and F are the scalar conic parameters obtained using an ellipse fitting algorithm [GGS94]. In order to remove the scale indeterminacy, the conics $\mathbf{C_1}$ and $\mathbf{C_2}$ are normalized to $\det \mathbf{C} = -1$. Extending the concept presented in [KO04], the normal vector to the target plane is given by:

$${}^{\mathcal{C}}\mathbf{n_0} = \mathbf{C_1} \begin{pmatrix} x_c/f_x \\ y_c/f_y \\ 1 \end{pmatrix} \tag{1.16}$$

where x_c and y_c are the pixel coordinates of the projected circle center. Finally, the center of circle $\mathbf{Q_1}$ in the camera frame, ${}^{\mathcal{C}}\mathbf{c_0}$ is computed as in [KO04]:

$${}^{\mathcal{C}}\mathbf{c_0} = \frac{\sqrt{\lambda^3}\, r_1\, (x_c/f_x,\, y_c/f_y,\, 1)^T}{{}^{\mathcal{C}}\mathbf{n_0}\, (x_c/f_x,\, y_c/f_y,\, 1)^T} \tag{1.17}$$

where λ is the smallest positive eigenvalue of $\mathbf{Q_1}$ and r_1 is the radius of the corresponding circle. The variables ${}^{\mathcal{C}}\mathbf{n_0}$ and ${}^{\mathcal{C}}\mathbf{c_0}$ represents an algebraic closed-form solution of the camera pose estimation which could be very sensible to noise conditions but usually a very good first estimation as shown in Fig. 1.11. Therefore, an accurate camera pose estimation is obtained by minimizing the conic parameters error obtained between the 3D circle image projection and the edged image of the conic:

$$f\left({}^{\mathcal{C}}\mathbf{n}, \left\|{}^{\mathcal{C}}\mathbf{c}\right\|\right) = \min\left(\epsilon\right) \tag{1.18}$$

where ϵ is the conic parameter error function:

$$\epsilon = \left\| e_x^2 + e_y^2 + e_M^2 + e_m^2 + e_\theta^2 \right\| \tag{1.19}$$

where e_x^2, e_y^2, e_M^2, e_m^2 and e_θ^2 are the errors of the ellipse center, the major axis, the minor axis and orientation angle respectively. It is worth to mention that the number of the nonlinear minimization parameters is reduced (i.e. 4 DOF) by taking into account the direction of the translation vector, ${}^{\mathcal{C}}\mathbf{c}$ (see Fig. 1.11). This vector is obtained by

$${}^{\mathcal{C}}\mathbf{c} = \mathbf{K}^{-1} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} \tag{1.20}$$

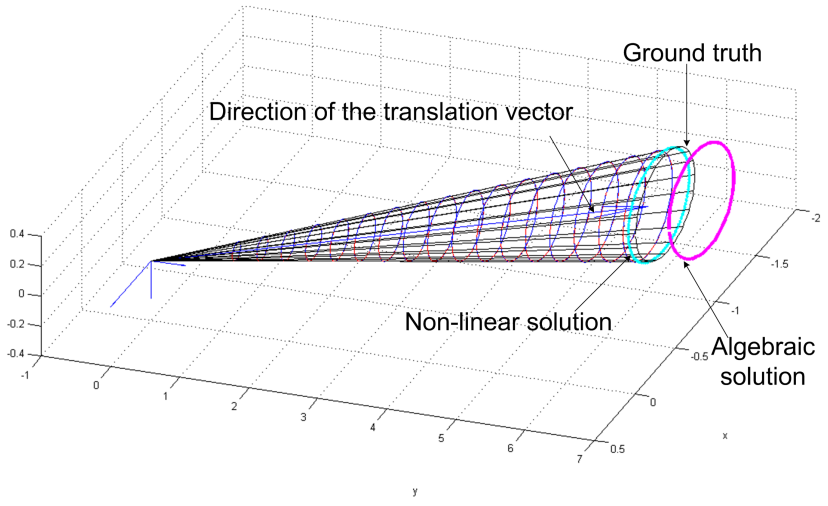


Figure 1.11: *Example of the camera pose estimation under an image noise of 3 pixels*

1.3.4 Lidar to Camera Transformation Estimation

A well-known closed-form solution for this problem is the method developed by [AHB87]. This method consists in obtaining the optimal rotation from the Singular Value Decomposition (SVD) of the correlation matrix of the centered point sets represented by $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = \left[{}^{\mathcal{L}}\hat{\mathbf{c}}_i - {}^{\mathcal{L}}\overline{\mathbf{c}}\right]_{3\times n}\left[{}^{\mathcal{C}}\hat{\mathbf{c}}_i - {}^{\mathcal{C}}\overline{\mathbf{c}}\right]_{n\times 3}^{T} = \mathbf{USV}^T \tag{1.21}$$

where n is the number of poses, ${}^{\mathcal{L}}\hat{\mathbf{c}}_i$ are the estimated coordinates of the 3D circle center from the i^{th} pose of lidar measurements. ${}^{\mathcal{L}}\overline{\mathbf{c}}$ is the centroid of the 3D-circle center point set in the lidar frame, ${}^{\mathcal{C}}\hat{\mathbf{c}}_i$ are the coordinates of the 3D-circle center point set estimated from the i^{th} pose of camera measurements and ${}^{\mathcal{C}}\overline{\mathbf{c}}$ is the centroid of the 3D-circle center point set in the camera frame. Therefore, the 3×3 optimal rotation matrix is obtained as follows:

$${}^{\mathcal{C}}\mathbf{R}_{0,\mathcal{L}} = \mathbf{VU}^T \tag{1.22}$$

where the subscript (0) denotes that it constitutes an initial estimation. The translation vector ${}^{\mathcal{C}}\mathbf{t}_{0,\mathcal{L}}$ is obtained as the vector which aligns the centroid of the 3D-circle center point set in the camera frame, ${}^{\mathcal{C}}\overline{\mathbf{c}}$, and the rotated centroid ${}^{\mathcal{C}}\mathbf{R}_{0,\mathcal{L}}.{}^{\mathcal{L}}\overline{\mathbf{c}}$:

$${}^{\mathcal{C}}\mathbf{t}_{0,\mathcal{L}} = {}^{\mathcal{C}}\overline{\mathbf{c}} - {}^{\mathcal{C}}\mathbf{R}_{0,\mathcal{L}}.{}^{\mathcal{L}}\overline{\mathbf{c}} \tag{1.23}$$

Non-linear solution

The rigid transformation obtained in the above section,${}^{\mathcal{C}}\left[\mathbf{R}_0,\, \mathbf{t}_0\right]_{\mathcal{L}}$ is a linear minimization of the Euclidean distance error between the 3D circle center point sets. This solution is usually a good starting guess of the extrinsic calibration. Therefore, in the aim of refining these estimated parameters, we first generate the 3D circles of the n poses estimated by the camera. It consists in computing m points of every estimated circle pose by using the 3D circle center and an orthonormal base lying in circle's plane. This orthonormal base is obtained from the normal vector to the circle's plane applying the Gram-Schmidt procedure [GvL96]. Let be ${}^{\mathcal{C}}\mathbf{p}_{i,k}$, the k^{th} generated 3D point using the camera estimation of the i^{th} pose. Secondly, the 3D circles of all the poses estimated by the lidar (section 1.3.2) are generated in the same way as presented for the camera estimations obtaining ${}^{\mathcal{L}}\mathbf{p}_{i,k}$. Then, the first guess is systematically applied for the rigid transformation, ${}^{\mathcal{C}}\left[\mathbf{R}_0,\, \mathbf{t}_0\right]_{\mathcal{L}}$ to get the points in the camera frame. Thirdly, under the assumption that the error orientation of the first guess rigid transformation is lower than $\pi/2$, we match the 3D points of the camera and lidar transformed estimations for every pose by using the nearest neighbor criterion as illustrated in Fig. 1.12. At this point, it is worth to mention that we have a 3D point set of camera and lidar observations associated. Finally, the refining of the rigid transformation parameters, $[\phi,\, \beta,\, \phi,\, t_x,\, t_y,\, t_z]^T$ is obtained by minimizing the following non-linear objective function:

$$\epsilon = \sum_{i=1}^{n}\sum_{k=1}^{m}\mathbf{W}.D_{ik}^2 \tag{1.24}$$

with

$$D_{ik} = \left\| {}^{\mathcal{C}}\mathbf{p}_{i,k} - {}^{\mathcal{C}}\mathbf{R}_{\mathcal{L},(\phi,\beta,\psi)}.{}^{\mathcal{L}}\mathbf{p}_{i,k} - {}^{\mathcal{C}}\mathbf{t}_{\mathcal{L},(t_x,t_y,t_z)} \right\| \tag{1.25}$$

where D_{ik} represents the Euclidean distance residual of the points after applying the rigid transformation and $\mathbf{W}$ is a weighting matrix. The results are obtained by using a robust M-estimator algorithm for calculating the robust weights as stated in [Ste99] and the LM-algorithm. After convergence, the solution of the calibration problem is represented by ${}^{\mathcal{C}}[\mathbf{R},\ \mathbf{t}]_{\mathcal{L}}$.

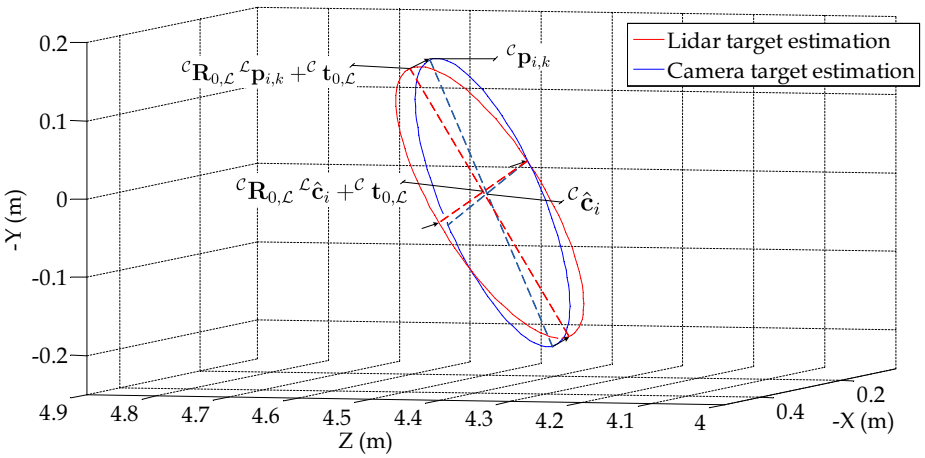


Figure 1.12: *Matching camera and lidar target pose estimations*

1.3.5 Calibration Accuracy Estimation

Thus far, we have estimated the rigid transformation between the camera and the ML lidar frame. The accuracy of the calibration results is estimated under the assumption that measurement errors are normally distributed. Therefore, the covariance matrix of the estimated parameters, $\mathbf{C}_\sigma$, is defined as follows :

$$\mathbf{C}_\sigma = MSE.\mathbf{J}.\mathbf{J}^T \tag{1.26}$$

with $MSE = \frac{1}{\theta-\varphi}\sum_{i=1}^{\theta}\epsilon^2$. $\mathbf{J}$ represents the Jacobian matrix of the last LM-algorithm iteration and MSE represents the mean squared error defined by θ, the number of observations, φ, the number of estimated parameters and ϵ the residual of the non-linear objective function. In our case, φ is equal to 6 (3 rotations and 3 translations) and $\theta - \varphi$ represents the degree of freedom of the χ^2 distribution. Based in the above classical approach for the covariance matrix of the non-linear fitted parameters, the width of the 95% confidence interval is obtained thanks to [Lup93]:

$$\delta\mathbf{C}_i = \sqrt{M_{\chi^2(95\%,\varphi)}}.\sqrt{\mathbf{C}_\sigma\,(i,\,i)} \tag{1.27}$$

where $\sqrt{\mathbf{C}_\sigma\,(i,\,i)}$ is the standard deviation of the estimated parameter and M the

corresponding value of the χ^2 distribution function.

1.3.6 Calibration Algorithm

Algorithm 1.3.1 Circle-based Lidar-Vision sensors Calibration Algorithm

Input: - Images of the circular pattern (for at least $n = 6$ poses)
- 3D range points of the circular pattern from lidar (for at least $n = 6$ poses)

Output: ${}^{\mathcal{C}}[\mathbf{R}, \mathbf{t}]_{\mathcal{L}}$

1: **for** $i = 1$ **do** n
2: ▶ Estimate the i^{th} lidar calibration pose, $\left[{}^{\mathcal{L}}\mathbf{n}, {}^{\mathcal{L}}\hat{\mathbf{c}}\right]_i$, as stated in section 1.3.2.
3: ▶ Estimate the i^{th} camera calibration pose, $\left[{}^{\mathcal{C}}\mathbf{n}, {}^{\mathcal{C}}\hat{\mathbf{c}}\right]_i$, as stated in section 1.3.3.
4: **end for**
5: ▶ Compute a first guess, ${}^{\mathcal{C}}[\mathbf{R}_0, \mathbf{t}_0]_{\mathcal{L}}$, for the lidar-camera rigid transformation using the linear solution (section 1.3.4)
6: **repeat**
7: ▶ Non-linear minimization using LM-algorithm according to the criterion Eq. 1.24
8: ▶ Robust noise variance estimation σ^2 based in non-linear minimization residuals
9: ▶ Weighting matrix $\mathbf{W}$ update
10: **until** convergence of ${}^{\mathcal{C}}[\mathbf{R}, \mathbf{t}]_{\mathcal{L}}$

1.4 Results

Experiments were carried out in simulated and in real conditions. Five simulations trials are reported hereafter, where the accuracy and consistency of the proposed calibration routine are respectively quantified and verified. Trials in real conditions using to different imaging sensors were performed achieving accurate results, crucial in further multi-modal experiments addressed in this study.

The calibration routine was implemented with MATLAB, since the procedure is an offline process which does not have real-time execution constraints.

1.4.1 Simulation Trials

Considering similar conditions to our multi-sensor system embedded, the simulation model correspond to the sensor relative position on board the vehicle.

The extrinsic parameters were set up to a constant translation vector in meters ${}^{\mathcal{C}}\mathbf{t}_{\mathcal{L}} = [-0.2, 0.8, 1.8]^T$ and an orientation matrix ${}^{\mathcal{C}}\mathbf{R}_{\mathcal{L},(\phi,\beta,\psi)}$, computed from the rotation angles $\phi = 11°$, $\beta = -1°$ and $\psi = 0.5°$ respectively in the X, Y and Z axis.

The ML lidar impacts were generated as the intersection of the lidar beam emission vectors and the simulated calibration target plane. A 3D space constraint was used

to ensure that all lidar layers scan the calibration object. A Gaussian white noise was added in the direction of the lidar beam emission simulating range measurement errors.

Based on the model stated in Eq. 1.5, a synthetic image was generated through a discrete image projection of the circle-based calibration target. The intrinsic parameters used in the camera model were a focal length in x direction f_x, a focal length in y direction f_y and principal point (u_0, v_0). Their numerical values were modified accordingly to the simulations test. A Gaussian white noise was added as well to the projected coordinates of the circle-based calibration target onto the image plane and to the intrinsic parameters of camera model.

Test No. 1: Influence of the number of poses

A first Monte Carlo-like simulation test was made in order to estimate the precision achieved by the proposed method using a minimal number of poses (worst case). To this end, six random poses were distributed and oriented randomly in the common field-of-view of the multi-sensor system by 100 trials. The noise added to the image coordinates and the focal lengths was fixed to a standard deviation of one pixel. Here, the intrinsic camera parameters were constrained to a unitary aspect ratio (i.e. $f_x = f_y = 1670$) and a principal point at the image center.

At each trial, the extrinsic parameters were estimated. The results obtained are presented in Table 1.1. Six poses seems to be not enough accurate considering that the obtained calibration results lead to a unacceptable image projection errors of objects located at the advertised range of the ML lidar (200m). Thus, a complementary test was performed in order to observed the error evolution as the number of poses changes.

Results of the Test No. 1 (Using only 6 poses)	
Relative Position error (millimeters)	46.1059
Relative Orientation error (degrees)	3.4362
Iterations	401

Table 1.1: *Achieved precision under a minimal number of poses*

As shown in Fig. 1.13, a considerable improvement is obtained as the number of poses increases. One can conclude that the use of seven to nine poses for calibration gives a good trade-off between accuracy and the number of poses to be acquired.

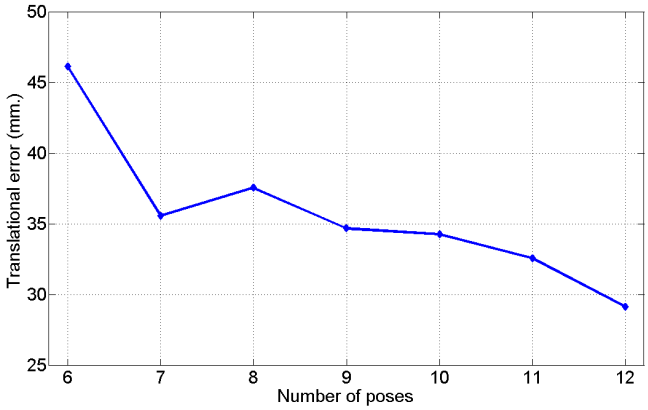
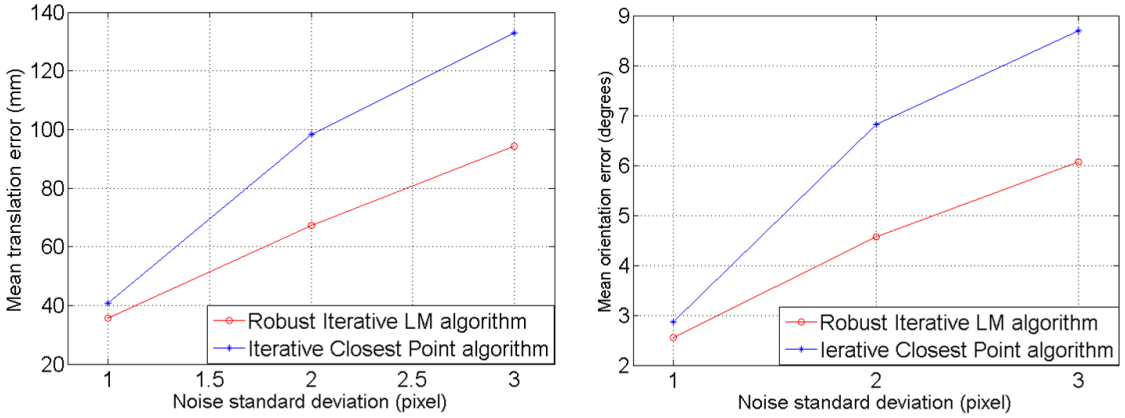


Figure 1.13: *Extrinsic calibration parameter error behavior by poses*

Test No. 2: Influence of image noise

In order for the error behavior of the method to be evaluated with respect to an image noise variation, a second simulation test was performed. In a similar way to the first simulation test, a hundred trials using seven poses were generated for each level of the image noise. The Gaussian white noise added spans between one to three pixels of standard deviation. Aiming at comparing the results obtained by the robust non-linear minimization of the 3D poses, we have also executed an Iterative Closest Point algorithm [BM92] as a reference of a classical registration of 3-D point sets. The results given in Fig. 1.14 evidences that the proposed robust registration scheme performs quite well and much better than the classical one.



(a) *Translational error (mm.)* **(b)** *Orientation error (degrees)*

Figure 1.14: *Extrinsic calibration errors regarding different image noise levels*

Test No. 3: Consistency test

The consistency of the estimated calibration accuracy, presented in section 1.3.5, is crucial for the error propagation and the uncertainty quantification when fusing lidar and vision data. As the estimated intervals ensure that the solution is included accord-

ingly to a given probability (e.g. $\pm 2\sigma$ for a 95% confidence), a consistency test was performed to verify this fact. To this end, the estimations of the rigid transformations and the confidence intervals using 7 poses in the calibration process were plotted over a hundred trials.

Fig. 1.15 illustrates as example the consistency test of the orientation angle parameter in the y-axis of the lidar frame (i.e. pitch vehicle angle) previously denoted β. A comparison of the results obtained through the Iterative Closest Point registration is shown as well. In the figure, the estimates were centered so as the ground truth value is the x plot axis.

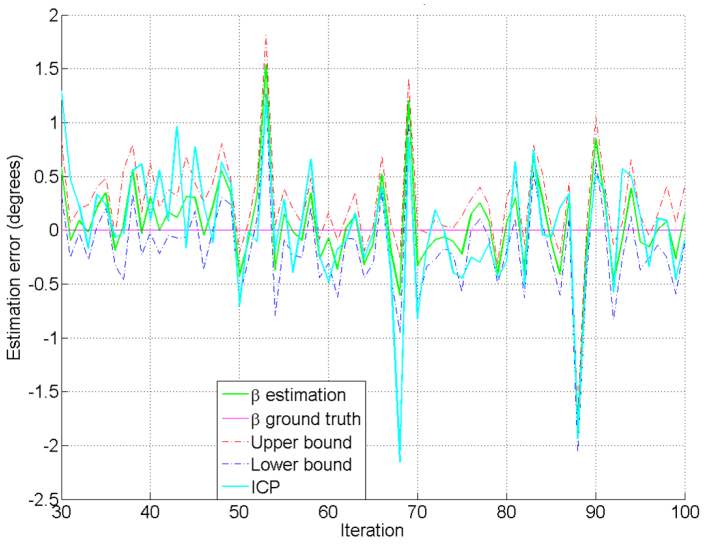


Figure 1.15: *Consistency test for β at 2σ*

One can notice that the robust non linear algorithm improves significantly the results and it has a good statistical efficiency but its convergence is not guaranteed as observed at the 53^{rd}, 69^{th} and 88^{th} trials.

Test No. 4: Influence of the projection camera model

Focusing now on the behavior of the extrinsic calibration method with regard to the projection camera model parameters, a new Monte Carlo-like test was performed with fifty trials each. In this test, seven random poses were used and a Gaussian-white noise which spans between one to three pixels of standard deviation was added to the image coordinates. The projection camera model used during the calibration process was constrained to a unitary aspect ratio (i.e. $f_x = f_y$), on the contrary the simulated model was not.

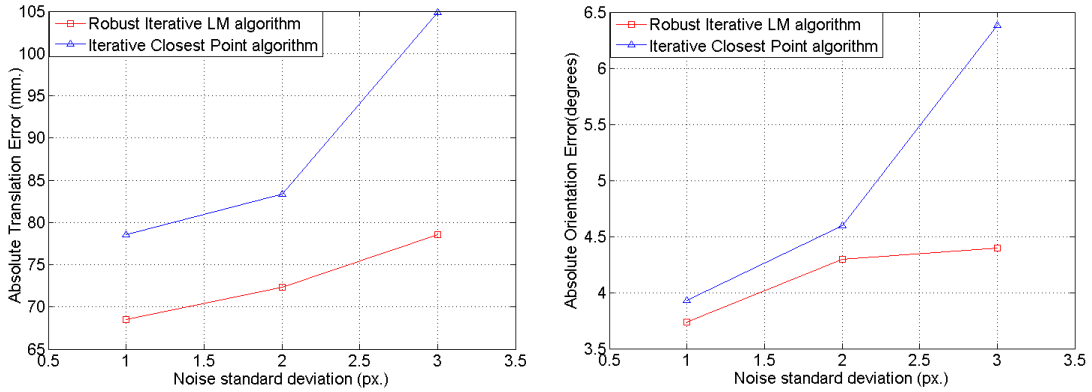


Figure 1.16: *Absolute errors using camera model with unitary aspect ratio: Translational (mm); Orientation (degrees)*

For every trial, the extrinsic parameters were estimated and the mean error of the fifty trials was computed. The results obtained by the robust non-linear minimization of the 3D poses are presented in Fig. 1.16. It can be noticed that even if the calibration parameters error is not negligible, it provides much better results than those obtained with the classical registration method.

A second test was performed taking into account the same conditions but now the camera model was unconstrained to a non-unitary aspect ratio. This test let us to estimate the influence of the camera model over the absolute error of the extrinsic calibration parameters. Fig. 1.17 illustrates an important improvement by using an unconstrained camera model.

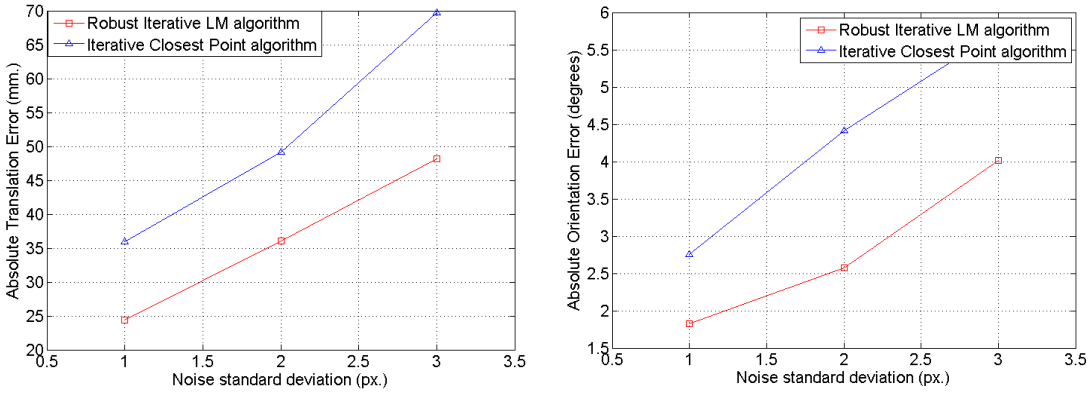


Figure 1.17: *Absolute errors using unconstrained camera model: Translational (mm); Orientation (degrees)*

The results of this last test reveal that the intrinsic parameters have an important influence over the extrinsic parameters. It is worth recalling that the extrinsic calibration method is based only on the minimization of the Euclidean error between the lidar and the camera perception. Therefore, errors in the intrinsic parameters will be compensated by the extrinsic ones so as to reduce this bias.

1.4.2 Real Experiments

The evaluation of the calibration method in real conditions was achieved between a ML lidar and two different vision systems: Monocular and stereo vision. In the sequel, the experimental condition details and the results for both considered configurations are reported.

Experiment using monocular vision

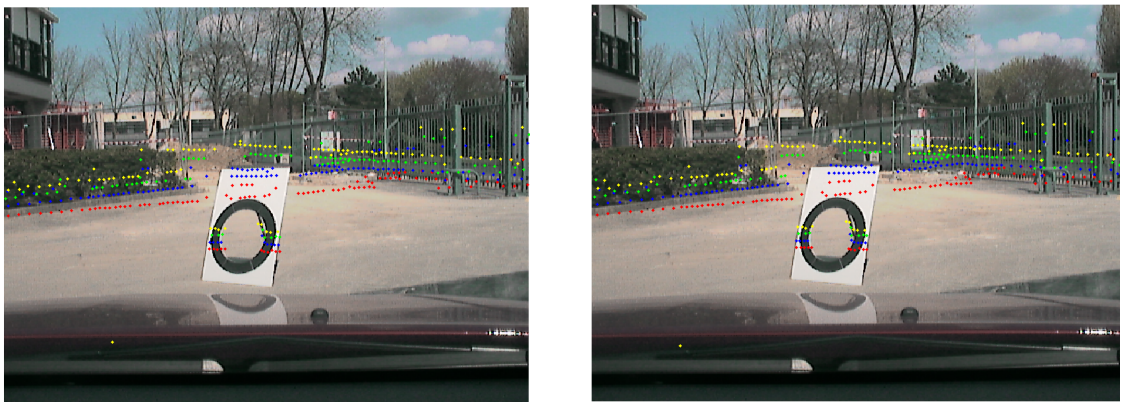
In this experiment an IBEO Alasca XT and a camera Sony DFW-VL500 were employed (see Fig. 1.18). The resolution of the camera was set to 640×480 pixels. We have used a calibration target with two concentric circles of radii 33cm and 23cm. The camera focal distances f_x and f_y have been estimated by using the circle-based calibration target. However, the principal point was estimated by using the classical Zhang's method [Zha00] due aux some instabilities observed under these experimental conditions (i.e. the important distance of the calibration target with respect to the camera).

A number of twenty scans were taken into account for each pose in the calibration process. Only 7 poses were used to estimate the initial guess solution for the rigid transformation.

Figure 1.18: *Experimental platform indicating the relative locations of the sensors*

Considering a unitary aspect ratio in the projection camera model, the extrinsic parameters were firstly estimated. By using this rigid transformation, a projection of the multi-layer measurements onto the scene image was achieved (see Fig. 1.19).

In a second trial, we used an unconstrained aspect ratio camera model and recomputed the corresponding extrinsic parameters.



(a) *Results using an unconstrained aspect ratio camera model*

(b) *Results using a unitary aspect ratio camera model*

Figure 1.19: *Projection image of lidar data using the extrinsic calibration method*

The image projection of the lidar data reveals that the error induced by the assumption of unitary aspect ratio in the projective camera model are almost imperceptible. In the Table 1.2, however, one can notice that the assumption provides an estimate of the relative sensors position which is biased. It was already noticeable by the increase in the absolute error in the previous simulations.

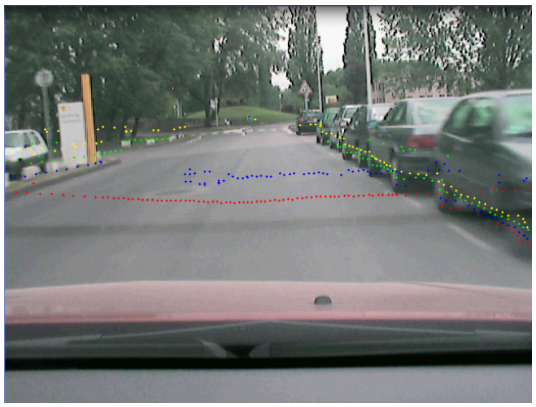
Results of the Test using Real Data					
Translation	Two Focal Lengths Camera Model	Confidence Interval	One Focal Length Camera Model	Confidence Interval	Measured
t_x	-0.1651 m	±0.0491	-0.3331 m	±0.0647	-0.2 m
t_y	0.9208 m	±0.1176	0.9246 m	±0.0648	0.88 m
t_z	1.8466 m	±0.0116	1.8027 m	±0.0087	1.82 m
Rot. angles					
$R_x = \phi$	1.5370 rad	±0.0296	1.5428 rad	±0.0156	n.a.
$R_y = \beta$	-0.0455 rad	±0.0118	-0.0505 rad	±0.0149	n.a.
$R_z = \psi$	1.6075 rad	±0.0466	1.6296 rad	±0.0261	n.a.

Table 1.2: *Results achieved with real data*

So as to provide additional evidence of the performance of our proposed approach, some snapshots acquired while driving, of the lidar data projected onto images, are illustrated in Fig. 1.20.

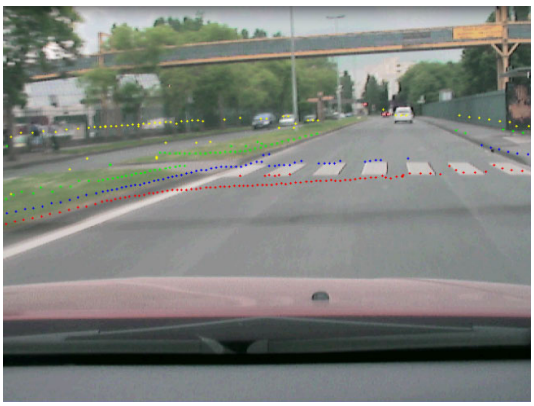
Experiment using stereo vision

In the ML lidar-stereo vision calibration trial, eight different target poses were acquired. Each pose was composed of twenty lidar scans and the corresponding target image of the left camera of the stereo rig. The stereo vision system was previously calibrated using a classical plane-based method [Zha00], hence their intrinsic parameters are estimated as well as the extrinsic ones between both cameras.



(a) *The correctness of the calibration parameters can be assessed in the geometrical coherence with the observed scene as shown in the detected parked vehicles.*

(b) *Even if the camera and the lidar data is received asynchronously, their slight temporal offset is almost imperceptible in dynamic objects as observed in the figure.*



(c) *Lidar impacts of objects located in a long range are projected correctly.*

Figure 1.20: *Image re-projection of multi-layer lidar data in dynamic scenes*

Then, the rigid transformation between the ML lidar and the left camera was computed and their corresponding intervals of confidence were estimated. The obtained relative orientation and position of the sensors are presented as an Euler vector in radians and a translation in meters respectively:

$$
\begin{aligned}
[\phi\ \beta\ \psi]^T &= \left[\begin{array}{ccc} 0.0874 \pm 0.0330 & -0.0327 \pm 0.0327 & 0.0114 \pm 0.0580 \end{array}\right]^T \\
[t_x\ t_y\ t_z]^T &= \left[\begin{array}{ccc} 0.3427 \pm 0.0953 & 1.0788 \pm 0.1055 & 2.5763 \pm 0.0236 \end{array}\right]^T
\end{aligned}
$$

Rotation parameters, $[\phi,\ \beta,\ \psi]^T$, are then converted into a quaternion formalizing the extrinsic parameters to the form ${}^{\mathcal{C}}[\mathbf{q},\ \mathbf{t}]_{\mathcal{L}}$. One can applied this transformation to the lidar point coordinates and then project them onto the left image. In order to project lidar scan points onto the right image, the lidar point coordinates in the left camera frame have to be transformed to the right camera one using the extrinsics of the stereo rig. Fig. 1.21 illustrates the resulting lidar data projection onto the stereo images.

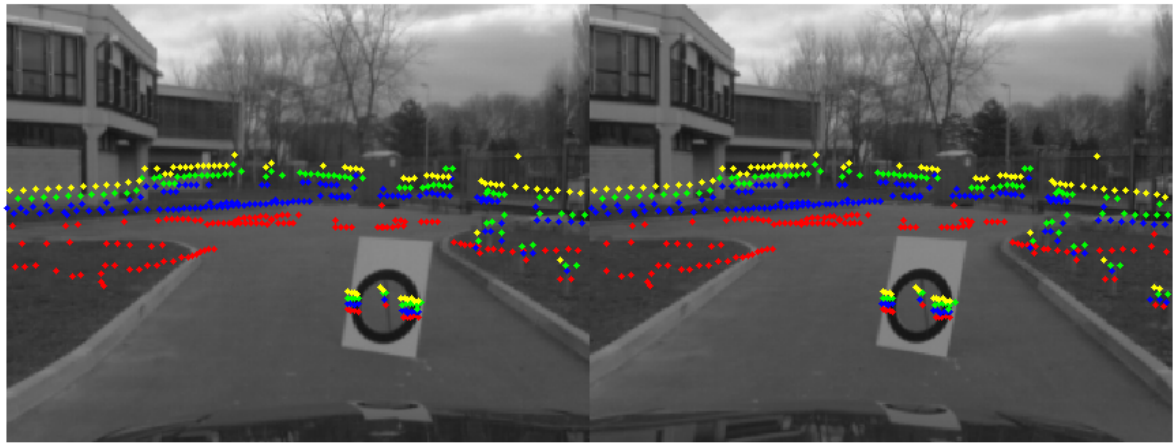


Figure 1.21: *ML Lidar data re-projection onto stereo images*

1.5 Conclusion

A new extrinsic calibration method for a common sensor configuration in vehicle applications has been proposed. By using a circle-based calibration target, extrinsic calibration and intrinsic camera calibration can be effectuated simultaneously. The results obtained in real data tests illustrate an appropriate and accurate projection of the lidar data. The estimation of the confidence intervals in the calibration method allows taking into account the error propagation in data sensor fusion applications. The integration of a full projection camera model can improve the calibration method. It constitutes a perspective of this research.

Chapter 2

Vision-based Odometry

Contents

2.1 Introduction

A reliable and precise estimation of the ego motion is essential for Advanced Driver Assistance Systems intended for safety applications, like obstacle collision avoidance systems. Such applications are usually designed to mitigate or avoid damages and in-

juries in particular conditions, for instance, in urban environments. Global Positioning System (GPS) is an affordable solution able to provide 3D positioning estimates (3D attitude is not available with one antenna) at an acceptable frequency (typically 10Hz). Unfortunately, its performance can be significantly decreased in urban environments because of multi-paths and satellites outages. *Dead-reckoning* is then a complementary solution which can be used when GPS information is unreliable. This localization technique relies on incremental integration of partial motion estimates to infer the vehicle pose starting from a known pose. A variety of techniques relaying on different perception means have contribute to solve the motion estimation problem. Among them, we can highlight wheel-speed sensor, inertial, range (e.g. lidar) and visual measurements strategies.

Problem Statement: Vehicle localization in a urban environment

In this chapter, the addressed problem consists in estimating the 3D pose of the vehicle as a function of time with respect to a static frame lying, for instance, in a tangent plane to the Earth surface. Hereafter, this problem is referred to the 3D ego-localization.

A previous work of Cappelle et al. [CNPC08] has shown that the 3D ego-localization of a vehicle can be achieved through a multi-sensor data fusion strategy of proprioceptive and exteroceptive sensors. Their proposed system operates using a GPS receiver, incremental encoders and a gyroscope. During degraded GPS conditions, the 3D vehicle pose is estimated using incremental encoders, a gyroscope, a camera and a 3D Geographical Information System (GIS).

Vision systems constitute nowadays affordable and promising environment sensing means for many on-board Intelligent Vehicle (IV) applications. Their large-spectrum perception capabilities are usually exploited for detection and recognition tasks, applied for instance, to roadsigns [KLD09], pedestrians [VJS03, GM06] and obstacles [Dic07]. However, cameras can be also suitable for 3D odometry, since they can provide estimates of the complete 6 degrees of freedom of a mobile platform (i.e. position and attitude) [vdMFDG02, Sol07, NNB06]. Agrawal et al. [AK06] have shown how visual odometry approaches can be complementary to the use of classical robotic techniques that rely on Inertial Measurement Units (IMU/Gyroscope) and Wheel Speed Sensors (WSS) which are subject to wheel slippage.

The motion estimation problem from images can vary depends deeply on the vision system configuration. For instance, the 3D trajectory of a monocular system can be retrieved from images over time, up to a scale factor (so called *Structure-from-motion problem*). Alternatively, Scaramuzza et al. [SFS09] have proposed an efficient real-time odometry method, based only on images provided by an omnidirectional camera. Stereo vision systems appear in this panorama of options, as an attractive sensing mean which grants access to the 3D structure of the scene, without any ambiguity, in a “snapshot”

way. Recently, Comport et al. [CMR07, CMR10] have proposed a novel 3D odometry method using a stereo vision system based on multiple view geometrical relations (i.e. 4-views or quadrifocal constraints). This approach makes use of a dense image warping technique which requires important optimization efforts and computational resources for real-time execution.

Our study focuses on the design and the implementation of a multi-modal localization strategy for vehicles immersed in urban environments. We focus here in the use of vision as dominant perception mean. This chapter is organized as follows: Section 2.2 presents how keypoints can be extracted to deal with relevant 3D scene information provided by images. This *sparse* approach is also crucial to satisfy real-time execution constraints. Section 2.3 presents a method intended to track observed keypoints over time. In this way, temporal scene analysis can be achieved by measuring the features apparent motion on image induced by the vehicle motion in space. In section 2.4, multiple view geometrical constraints are studied in order to predict keypoints locations given 3D a motion of the vision system.

By combining all of these theoretical principles, an ego-motion estimation method is proposed to address the vehicle localization problem. This technique provides a positive balance between precision and execution time thanks to a sparse feature strategy.

In a first vision-only based approach (see section 2.5), the algorithm finds the ego-motion parameters which minimize the error between the measured and the predicted motion of a feature set, in the image plane. The measured feature image motion is obtained using a pyramidal feature tracking. The predicted motion is computed as a quadrifocal warping, based on multiple view geometrical constraints.

A second strategy is proposed using stereo vision and WSS-Gyro fused with a tightly coupled scheme (see section 2.6). This technique copes with each perception drawback, increasing the precision and the reliability of the perception system.

The remaining of the chapter is dedicated to cover the computation of the vehicle position by integrating ego-motion estimates. The problem of obtaining a Geo-reference positioning is also presented. Finally, results in simulated and real-time execution conditions are reported and analyzed.

2.2 Features Extraction

Dealing with video sources involves quite an important amount of data. Processing all image data in a dense way is a time consuming task. A usual method to reduce image processing time is to determine what is interesting in the scene by its observed motion. For this, unidentified objects observed in a scene can be visually tracked based on key local features rather than complex representations.

An image keypoint is a local feature which is near to be unique and can be compared to other keypoints based on a set of parameters. Different kinds of image keypoints (i.e. features) are known in the literature. Harris, SIFT (Scale Invariant Features Transform) and SURF (Speeded-Up Robust Features) features are part of a representative but non exhaustive list of image keypoints.

This section is intended to present different techniques for extracting relevant photometric information from images by means of keypoint detectors.

Harris Features

Attempting to solve the keypoint association and tracking between images, Harris et al. [HS88] have proposed a corner and edge detector based on a local autocorrelation function. In this way, corner, edge and flat regions are efficiently detected using an anisotropic correlation response, $f(du, dv)$, induced by a small shift, (du, dv), as stated in the following:

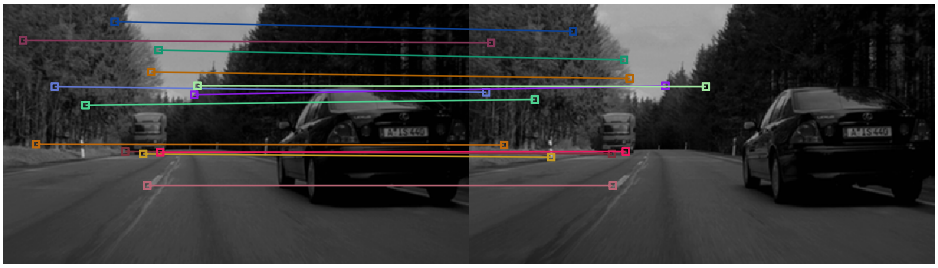
$$f(du, dv) \approx [du \;\; dv]\mathbf{M}[du \;\; dv]^T \quad \text{with} \quad \mathbf{M} = \sum_{u,v} \mathbf{W}(u, v) \begin{bmatrix} \mathbf{I}_{uu} & \mathbf{I}_{uv} \\ \mathbf{I}_{uv} & \mathbf{I}_{vv} \end{bmatrix} \tag{2.1}$$

where the shape of the autocorrelation function f at the origin is denoted by $\mathbf{M}$, the image pixel intensities are represented by $\mathbf{I}$, the image coordinates are $(u\, v)$ and the partial derivatives of $\mathbf{I}$ in the u and v directions are respectively $\mathbf{I}_u$ and $\mathbf{I}_v$. $\mathbf{W}$ is a windowing function (e.g. Gaussian weighting matrix). The Harris key features are parametrized by the rotational invariant eigenvalues of $\mathbf{M}$ and the corner response $r = \mathrm{Det}(\mathbf{M}) - k\,\mathrm{Tr}(\mathbf{M})^2$ with k as a constant.

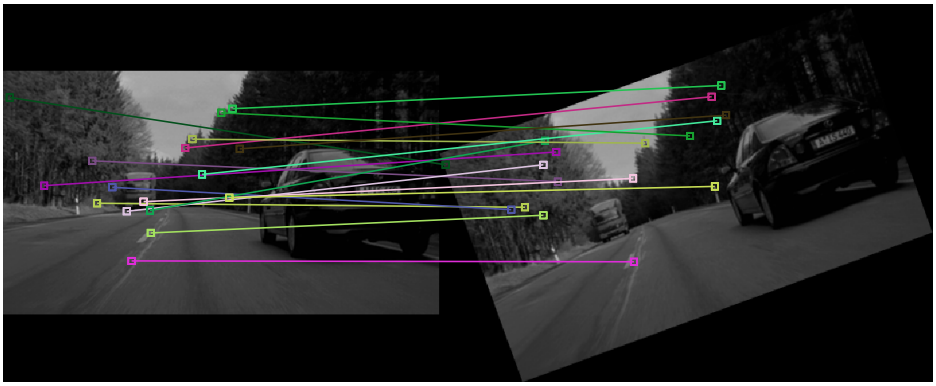
Despite the computational effectiveness of the corner feature detector, it has some limitations like its low repeatability rate when scale and affine changes are involved as illustrated in Fig. 2.1. Other keypoints have been proposed to overcome these limitations like SIFT and SURF features.

Scale-Invariant Feature Transform Features

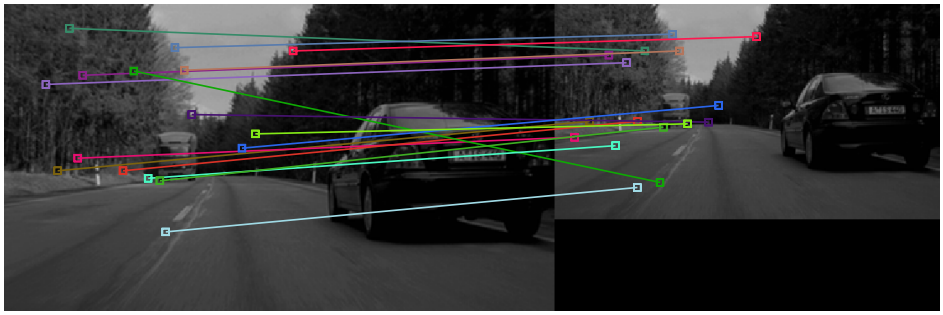
The Scale-Invariant Feature Transform [Low04] provides local keypoints which are characterized by highly distinctive descriptors. The fundamental principle of this feature extraction approach is based on a keypoint detection which is performed in the spatial and frequency domains. This makes the approach robust to image scaling, rotation and partially invariant to illumination and 3D camera viewpoint changes. The method can be outlined in four major stages applied in a cascade filter algorithm: scale-space extrema detection, accurate keypoint localization, orientation assignment and keypoint descriptor computation.



(a) *The example illustrates the distinctiveness and repeatability of Harris features in a lateral translation change*



(b) *The example shows that Harris features are not robust to rotation changes*



(c) *Scale changes represent another limitation in the use of Harris features as shown here*

Figure 2.1: *Harris feature detection on a road scene. In this example some Harris keypoints were extracted and their image positions were denoted by color squares. Their repeatability and distinctiveness is illustrated through the identification of same features after translation, rotation and scale image changes.* [1]

Firstly, the keypoints candidates are detected as the maxima/minima of a multi-scale Difference of Gaussians (DoG). For this, the image is convolved at different scales with different Gaussian filters. Then, the DoG is obtained as the successive difference between the convolved images. The keypoint candidates are then locally identified across the scales as the extrema in the DoG as illustrated in the Fig. 2.2.

[1] Original image source from database of the MOBILE3DTV project funded by the Europe's Seventh Framework Programme

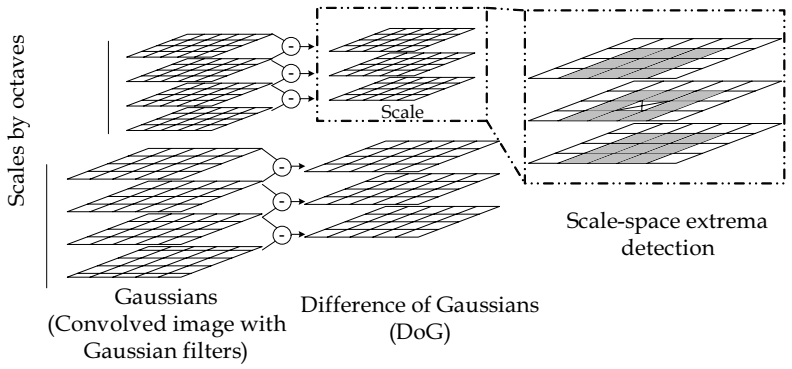


Figure 2.2: *Scale-space extrema detection- This figure is as schematic of the Scale-space extrema detection*

Secondly, keypoint candidates are localized accurately and those with low-contrast or issued of poor edge responses are rejected. This step attempts to ensure the stability of the keypoints. The accurate location of a keypoint candidate is estimated by interpolating its position using the quadratic Taylor expansion of the DoG space-scale function, $\mathbf{D}(\mathbf{x})$, such as,

$$\mathbf{D}(\mathbf{x}) = \mathbf{D}(\mathbf{0}) + \left(\frac{\partial \mathbf{D}^T}{\partial \mathbf{x}}\right)_{\mathbf{x}=\mathbf{0}} \mathbf{x} + \frac{1}{2}\mathbf{x}^T \left(\frac{\partial^2 \mathbf{D}}{\partial \mathbf{x}^2}\right)_{\mathbf{x}=\mathbf{0}} \mathbf{x} \tag{2.2}$$

where $\mathbf{D}$ has been shifted so that the keypoint location, $(u,\, v,\, \sigma)$, is placed at the origin and the location offset $\mathbf{x} = [du\, dv\, d\sigma]^T$ is defined in the space (u, v) and the scale (σ). The precise offset, $\hat{\mathbf{x}}$, from the location of the keypoint candidate is finally determined by setting to zero the derivative of $\mathbf{D}(\mathbf{x})$ with respect to $\mathbf{x}$. Once $\hat{\mathbf{x}}$ has been estimated, the keypoint candidate is kept according to the following criteria:

- If $\hat{\mathbf{x}} > 0.5$, the keypoint candidate is discarded since there exists another keypoint closer to the image feature
- If $\hat{\mathbf{x}} < 0.5$, the candidate is kept and the final keypoint location is obtained by shifting its position by this offset. In order to avoid low-contrast keypoints, those having $|\mathbf{D}(\hat{\mathbf{x}})| < 0.03$ are rejected.

Then, the keypoints issued of poor edge responses are identified and hence rejected based on the eigenvalues of the second-order Hessian matrix, $\mathbf{H}$, following a quite similar approach to the one suggested in the Harris corner detector:

$$\mathbf{H} = \begin{bmatrix} \mathbf{D}_{uu} & \mathbf{D}_{uv} \\ \mathbf{D}_{uv} & \mathbf{D}_{vv} \end{bmatrix}$$

Knowing that the eigenvalues of $\mathbf{H}$ are proportional to the principal curvatures of $\mathbf{D}$, the following quality test is performed:

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

where r is an appropriate ratio threshold (e.g. $r = 10$) .

Thirdly, the orientation of every keypoint is computed and assigned to the feature descriptor, achieving in this manner robustness to rotational changes. For this, the local gradient magnitude, $g(u, v)$, and orientation, $\theta(u, v)$, are computed using pixel differences in the Gaussian smoothed image, $\mathbf{L}$, which corresponds to the nearest keypoint scale:

$$g(u,v) = \sqrt{(\mathbf{L}(u+1,v) - \mathbf{L}(u-1,v))^2 + (\mathbf{L}(u,v+1) - \mathbf{L}(u,v-1))^2} \quad (2.3)$$

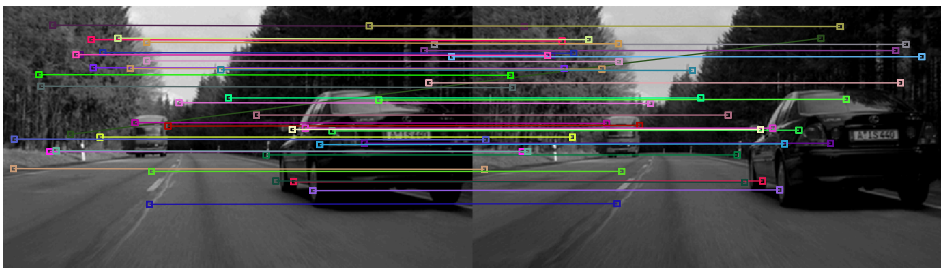
$$\theta(u,v) = \tan^{-1}\left((\mathbf{L}(u,v+1) - \mathbf{L}(u,v-1)) \,/\, (\mathbf{L}(u+1,v) - \mathbf{L}(u-1,v))\right) \quad (2.4)$$

Subsequently, an orientation histogram of 36 discretized local gradient orientations (i.e. bins representing 10 degrees intervals) of the keypoint is computed. The highest histogram peak creates a SIFT keypoint with that orientation. In the case where other histogram peaks are withing 80% of the highest one, multiple SIFT keypoints are created with the corresponding orientations.

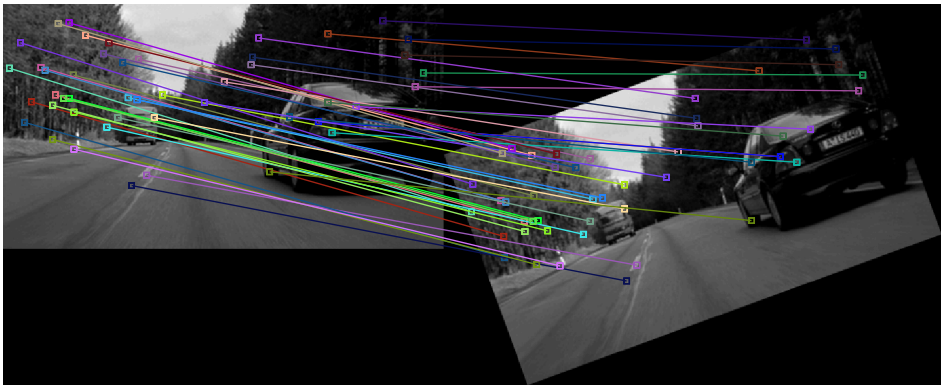
The previous steps provide a set of SIFT keypoints characterized by their location, scale and orientation ensuring invariance to rotational and scaling changes. The fourth and last step consists in computing a distinctive descriptor composed of histograms which are characterized by magnitude and orientation. Arranged in a normalized 128-vector (8 orientation bins of 4x4 location bins), the SIFT keypoints enhance the invariance to minor affine and illumination changes. Fig. 2.3 shows the SIFT feature distinctiveness invariance in translation, rotation and scale changes, in a road scene.

Speeded-Up Robust Features

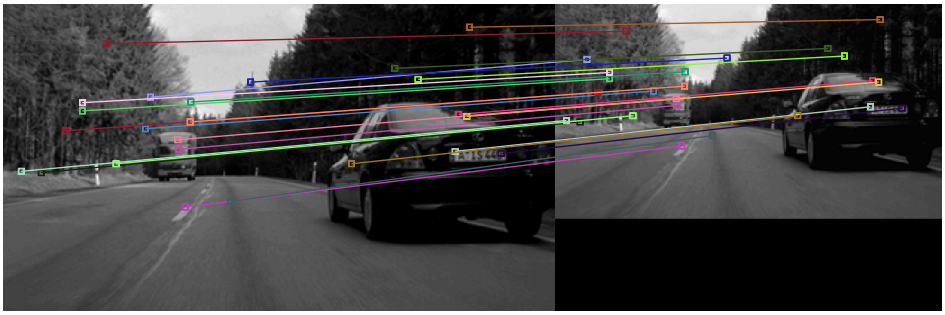
In [BETG08], Bay et al. have proposed a new approach for local keypoint extraction, description and matching, called Speeded-Up Robust Features, SURF for short, which are distinctive and relatively fast for on-line applications. The keypoint extraction algorithm follows the basic scheme of the SIFT features with various refinements. One of these relevant enhancements is the use of the integral image concept in order to drastically decrease the computation time, particularly for the image-Gaussian filters convolution.



(a) *SIFT Features are quite robust to translation image changes as illustrated in the figure where just one mistaken identification*



(b) *In this example, the use of the SIFT Feature descriptors provides a reliable feature identification in rotation changes*



(c) *The performance of the SIFT features identification remains invariant in scale image changes*

Figure 2.3: *SIFT feature detection on a road scene. In this example some a set of SIFT keypoints were extracted in a road scene. The invariance of the SIFT features to translation, rotation and scale changes is demonstrated through the correctness of the feature identification.*

SURF keypoints candidates are localized across scales based on a space-scale representation obtained by the use of the convolution of the second order derivatives Gaussian filters with the image. For this, the Hessian matrix and its determinant are efficiently approximated for a given point $\mathbf{x}_{(u,v)}$ at scale σ using Haar wavelets (i.e. Fast-Hessian):

$$\mathbf{H} = \begin{bmatrix} \mathbf{D}_{uu}(\mathbf{x},\sigma) & \mathbf{D}_{uv}(\mathbf{x},\sigma) \\ \mathbf{D}_{uv}(\mathbf{x},\sigma) & \mathbf{D}_{vv}(\mathbf{x},\sigma) \end{bmatrix} \tag{2.5}$$

$$\mathrm{Det}(\mathbf{H}) = \mathbf{D}_{uu}\mathbf{D}_{vv} - (w\mathbf{D}_{uv})^2 \tag{2.6}$$

where w is a regularization value which changes depending on the scale, $\mathbf{D}_{uu}$, $\mathbf{D}_{vv}$ and $\mathbf{D}_{uv}$ correspond to the second order Gaussian partial derivatives in u, v, and uv directions respectively. The locations of the SURF keypoints are estimated by a maximum detection in the space-scale neighborhood (i.e. u, v, σ) and are then refined by interpolation.

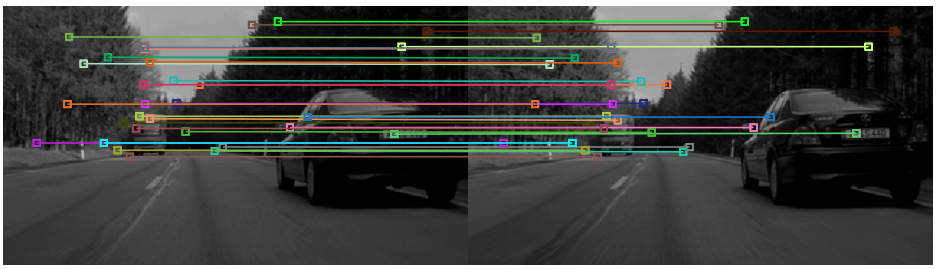
Once the SURF keypoints have been localized, each point is described by its orientation and descriptor, similarly to the concept of the SIFT keypoints. The orientation of the keypoint is estimated as the sum of the horizontal and vertical Haar wavelets responses, obtained in the circular surrounding region of the point. The SURF descriptor is estimated in a square region oriented as the keypoint. For this, the sum of the horizontal and vertical Haar wavelet responses and the sum of their absolute values are computed in 4×4 sub-regions. Finally, the obtained values are concatenated in a normalized 64-vector (i.e. unit vector) achieving invariance to contrast.

Applying the presented SURF feature detector in a road scene image, it can be remarked in Fig. 2.4, that this detector is as robust as SIFT features. The extraction of SURF features requires, however, a smaller computation time than the SIFT detector. A summary of key characteristics of the presented feature point detectors is provided in Table 2.1.

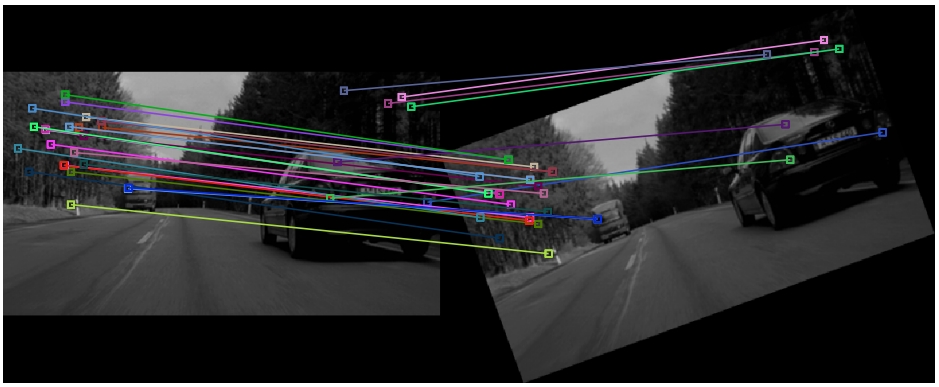
Table 2.1: *Feature detectors summary*

Feature Points	*Harris*	*SIFT*	*SURF*
Repeatability	•	••	••
Rotation, Affine and Scale changes invariance	–	•••	••
Illumination changes	–	••	••
On-line applications	•••	•	••

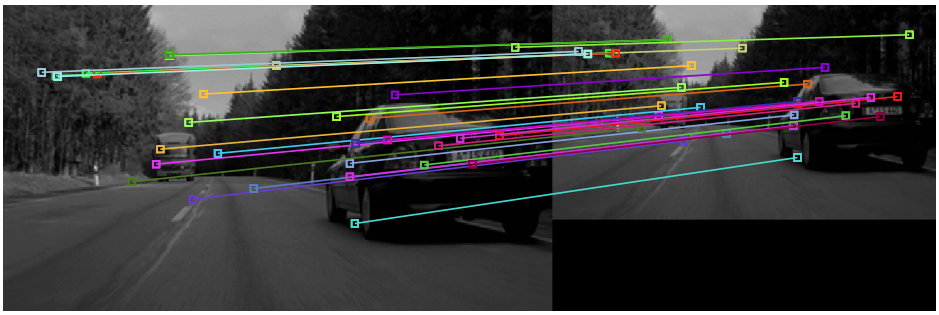
In practice, Harris and SURF keypoints are well-adapted features for on-line vehicle applications, since they are distinguishable enough in road scenes and their detection performs quite fast. As it can be noted in Fig. 2.1, 2.3 and 2.4, features are rarely detected in the image section corresponding to the road, because it constitutes a homogeneous texture. Such situations can lead to obtaining keypoints located only in a small image region. A strategy attempting to get a well distributed set of keypoints, consist in partitioning the image by sectors (so called *bucketing* feature detection). Then, a constant number of features by sector is extracted, ensuring scattered keypoints in all the image area.



(a) *SURF features robustness in translation image changes*



(b) *SURF features are invariant to rotation image changes*



(c) *The SURF features performance in scale changes is illustrated in this example*

Figure 2.4: *SURF feature detection on a road scene. In this example SURF keypoints were extracted and its descriptor invariance was tested regarding translation, rotation and scale changes.*

2.3 Feature Tracking

Thus far, the 3D information provided by a vision system has been considered as motionless. However, processing consecutive images over time can also provide helpful information for scene understanding and 3D motion estimation. Therefore, our attention is now aimed at measuring the apparent motion of the unidentified objects observed in the image plane. This apparent motion estimation (also known as *optical flow*) can be computed through a sparse (also known as *feature tracking*) or a dense tracking of the "local pixel displacements". For more details about dense tracking please refers to [HS81].

The feature tracking problem consists in determining the image displacement of a

keypoint based on a local brightness similarity error as proposed by Lucas and Kanade in [LK81]. For this basic idea, three assumptions must be considered (see Fig. 2.5). The first one is brightness constancy, which assumes that the brightness of the key feature does not change as it moves from image to image. The second assumption considers an infinitesimal increment of time between frames ensuring small image motions. The third is the spatial coherence, which assumes that the surrounding pixels of a keypoint belong to the same surface with quite similar observed motions.

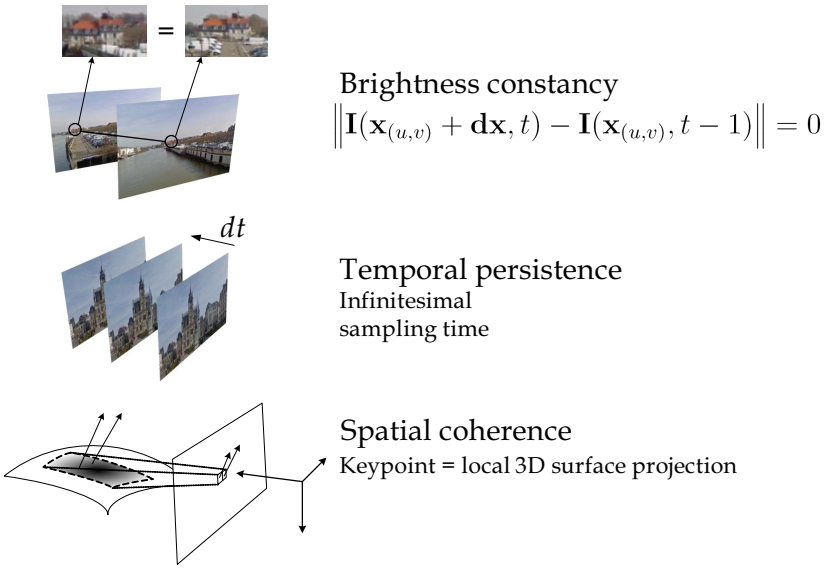


Figure 2.5: *Feature Tracking Assumptions. The brightness constancy, which assumes that the brightness of the key feature does not change over time. The second assumption considers small image motions between frames. The spatial coherence assumes that the surrounding pixels of a keypoint belong to the same surface with quite similar observed motions.*

Based on the above assumptions, the frame-to-frame motion of a key feature is estimated by locally minimizing (i.e. correlation window) the brightness error between the observed keypoint at two instants of time,

$$\epsilon = \sum_{\mathbf{x}} \Big[\mathbf{I}(\mathbf{w}(\mathbf{x}_{(u,v)},\mathbf{dx}),t) - \mathbf{I}(\mathbf{x}_{(u,v)},t-1)\Big]^2 \tag{2.7}$$

where $\mathbf{I}(\cdot,t-1)$ and $\mathbf{I}(\cdot,t)$ are the local image templates of the observed keypoint at consecutive times (for simplicity, time is indexed as $t-1$ and t, respectively), $\mathbf{dx}$ represents the vector of parameters to be estimated, $\mathbf{x}_{(u,v)}$ denotes the image coordinates of the tracked keypoint $\mathbf{x}$ and $\mathbf{w}(\cdot)$ is the warping operator which models the apparent motion parametrized by $\mathbf{dx}$. This operator associates the templates $\mathbf{I}(\cdot,t-1)$ and $\mathbf{I}(\cdot,t)$ for a given parameter $\mathbf{dx}$. In [BAHH92], an affine warping operator was proposed for tracking an image patch moving in 3D. For computing the apparent motion of the tracked keypoints, it can be considered a warping operator representing a 2D image translational motion defined as [BM04, ST94]:

$$\mathbf{w}(\mathbf{x}_{(u,v)}, \mathbf{dx}) = \mathbf{x}_{(u,v)} + \mathbf{dx} = \begin{bmatrix} u + du \\ v + dv \end{bmatrix} \tag{2.8}$$

where $\mathbf{dx} = \mathbf{u}\, dt = [du,\, dv]^T$ is the 2D image displacement of the keypoint. Note that $\mathbf{u}$ represents some kind of velocity of the keypoint in the image.

Thus, Eq. 2.7 yields,

$$\epsilon = \sum_{\mathbf{x}} \Big[\mathbf{I}(\mathbf{x}_{(u,v)} + \mathbf{dx}, t) - \mathbf{I}(\mathbf{x}_{(u,v)}, t-1)\Big]^2 \tag{2.9}$$

The quadratic non-linear function $\epsilon(\mathbf{u})$ can be then formulated as a minimization problem that can be solved iteratively based on the gradient constraint and starting at initial conditions $\mathbf{u}_0$. For this, $\epsilon(\mathbf{u})$ can be rewritten using the first order Taylor expansion,

$$\epsilon(\mathbf{u}) = \sum_{\mathbf{x}} \Big[\boldsymbol{\nabla}\mathbf{I}^T\mathbf{u} + \dot{\mathbf{I}}\Big]^2 \tag{2.10}$$

with $\quad \dot{\mathbf{I}} = \mathbf{I}(\mathbf{x}_{(u,v)} + \mathbf{dx}, t) - \mathbf{I}(\mathbf{x}_{(u,v)}, t-1)$

where $\mathbf{u} = \left[\frac{du}{dt},\, \frac{dv}{dt}\right]^T$ is the *optical flow*. $\dot{\mathbf{I}}$ and $\boldsymbol{\nabla}\mathbf{I}$ represents the temporal and spatial image derivatives evaluated at $\mathbf{x}_{(u,v)}$. Note that these derivatives are directly measurable from images.

Then, for each keypoint, the optimal values of $\mathbf{u}$ which minimize $\epsilon(\mathbf{u})$ can be obtained at each iteration as the least squares estimate of $\mathbf{u}$ which satisfies:

$$\nabla\epsilon(\mathbf{u}) = 2\sum_{\mathbf{x}} \boldsymbol{\nabla}\mathbf{I}\Big[\boldsymbol{\nabla}\mathbf{I}^T\mathbf{u} + \dot{\mathbf{I}}\Big] = 0 \tag{2.11}$$

In this way, the estimate of $\mathbf{u}$ is updated until either the brightness similarity error becomes minimal or the maximum number of iteration is reached.

2.3.1 Aperture problem

One of the constraints considered in the feature tracking problem is the brightness constancy formalized in Eq. 2.10. Recalling this assumption, a tracked keypoint must satisfies:

$$\boldsymbol{\nabla}\mathbf{I}^T\mathbf{u} + \dot{\mathbf{I}} = 0 \tag{2.12}$$

Since the brightness constancy equation only provides a single constraint for the two components of the optical flow, $\mathbf{u} = \left[\frac{du}{dt},\, \frac{dv}{dt}\right]^T$, Eq. 2.12 could be singular for keypoints lying on edges. In consequence, only the component of the *optical flow* which is normal to the image intensity gradient (called *normal flow*) can be determined. This restriction is known as the aperture problem and is easily visualized in contours where the

motion along lines (called *tangential flow*) is indiscernible in the correlation window as illustrated in Fig. 2.6.

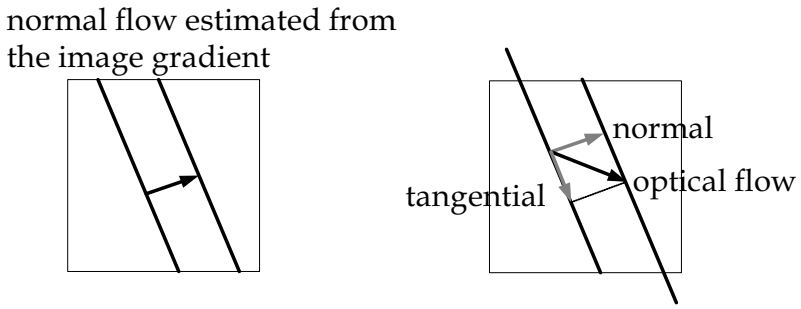


Figure 2.6: *Aperture problem restricts the computed apparent motion to the normal flow.*

2.3.2 Variants of the feature tracking

The Lucas-Kanade feature tracking algorithm offers an efficient method to estimate the apparent motion. However, its performance is limited essentially by the approach assumptions. For instance, the estimated image displacements are quite restricted (i.e. up to 1 pixel) due to the "*small motion*" assumption. For this, a coarse-to-fine optical flow estimation strategy has been proposed by Bouguet in [Bou02] achieving in this way robustness to the assumption violation.

Thus, a pyramid constituted of the sub-sampled image at different scales allows the iterative estimation of the optical flow (see Fig. 2.7). A coarse optical flow is estimated at the lowest image resolution, tracking large image motions, and is then propagated and refined up to the maximal image resolution, to track small motions.

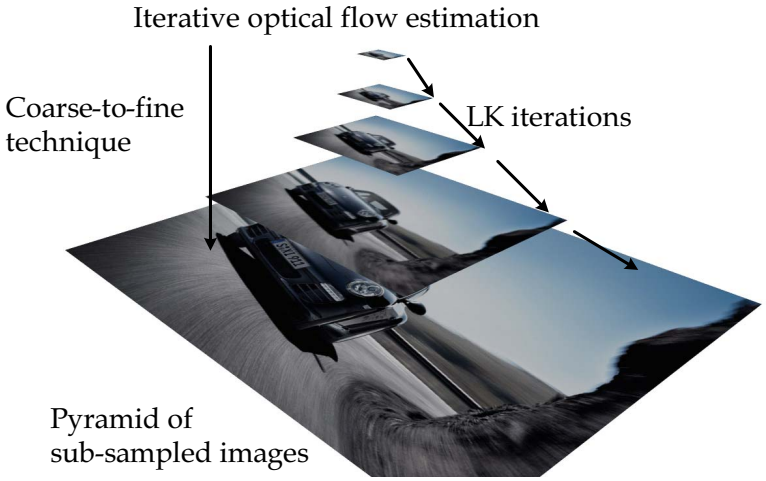


Figure 2.7: *The image pyramid structure allows the tracking of important inter-frame feature motions relaxing in this way the temporal persistence assumption.*

2.4 Multiple View Geometry Constraints

2.4.1 Two-view geometry

Epipolar geometry and fundamental matrix

Consider two different views of a scene provided by calibrated cameras, rigidly linked by a baseline axis. The engendered geometry of these two views, is defined by the family of planes passing through the baseline axis and the image planes as illustrated in Fig. 2.8. This geometry, known as *epipolar geometry*, is completely invariant to the scene structure and depends only on the relative pose between the cameras (i.e. extrinsic) and their internal parameters (i.e. intrinsic).

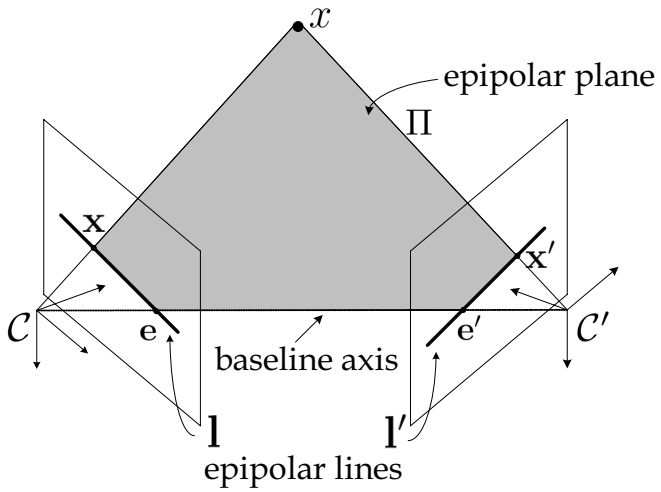


Figure 2.8: *Epipolar Geometry engendered by two views*

Let x be a point in the 3D space of the scene which is projected, respectively in both views, at $\mathbf{x} = [u, v, 1]^T$ and $\mathbf{x}' = [u', v', 1]^T$ representing its corresponding homogeneous image coordinates. A plane Π passing through x and the camera centers can be defined. This projective plane establishes a geometrical link between both views, since $\mathbf{x}$ and $\mathbf{x}'$ are coplanar and lie in the epipolar plane Π. Additionally, this geometrical link introduces and defines new entities: epipoles and epipolar lines. The epipoles are the imaged points where the baseline axis intersects with image planes in the left view, noted $\mathbf{e}$, and the right noted $\mathbf{e}'$. The epipolar lines, noted $\mathbf{l}$ and $\mathbf{l}'$, are the intersections between the images and the epipolar plane as illustrated in Fig. 2.8.

We now focused on a direct mapping which allows us to determine the epipolar line, $\mathbf{l}'$, corresponding to each point $\mathbf{x}$ in the left view. This mapping will constrain the two-view feature matching to a simple search of the corresponding point $\mathbf{x}'$ along the epipolar line.

The *fundamental matrix*, noted $\mathbf{F}$, formalizes this mapping by encapsulating the epipolar geometry in an algebraic representation. Since this mapping transfers from a 2-

dimensional onto a 1-dimensional projective space, it is defined by a 3×3-matrix of rank 2.

As depicted in Fig. 2.9, the fundamental matrix can be inferred geometrically by a point transfer through a plane Ω [HZ03]. For this, a 2D homography, $\mathbf{H}_\Omega$ is computed, since the correspondence of a set of imaged points, $\mathbf{x}_i$ and $\mathbf{x'}_i$, is known. These imaged points are issued of the projection of a set of points x_i lying in Ω. Then, the epipolar line in the second view $\mathbf{l'}$, passing through the epipole $\mathbf{e'}$ and $\mathbf{x'}$ can be defined by the following crossed product:

$$\mathbf{l'} = \mathbf{e'} \times \mathbf{x'} \tag{2.13}$$

where $\mathbf{x'}$ can be conveniently replaced by $\mathbf{H}_\Omega \mathbf{x}$. Hence,

$$\mathbf{l'} = [\mathbf{e'}]_\times \mathbf{H}_\Omega \mathbf{x} = \mathbf{F}\mathbf{x} \tag{2.14}$$

where the operator $[\cdot]_\times$ denotes the skew-symmetric matrix form.

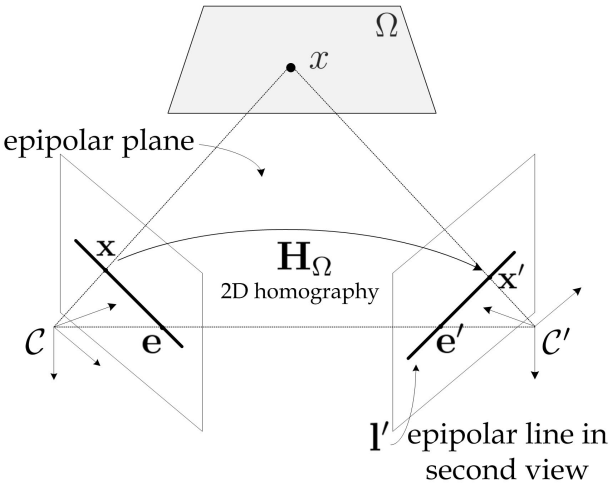


Figure 2.9: *Geometrical inference of the Fundamental Matrix by a point transfer through a plane.*

Alternatively, the epipolar line in the second view can be obtained in an algebraic manner. For this, consider two camera projection matrices referenced with respect to the left camera. The left and right canonical projective camera matrices are noted $\mathbf{P}_0$ and $\mathbf{P'}_0$ respectively. They are defined as follows:

$$\mathbf{P}_0 = \mathbf{K}\,[\mathbb{I}_{3\times 3}\,\mathbf{0}] \tag{2.15}$$

$$\mathbf{P}'_0 = \mathbf{K}'\,[\mathbf{R}\,\mathbf{t}] \tag{2.16}$$

where $\mathbb{I}_{3\times 3}$ is the identity matrix, $\mathbf{K}$ and $\mathbf{K}'$ are 3×3 matrices containing the intrinsic camera parameters and $[\mathbf{R}\,\mathbf{t}]$ are the extrinsic parameters expressing the orientation and the position of the right camera with respect to the left one. In this manner, the

back-projection of the imaged point $\mathbf{x}$ up to a scale factor λ is given by:

$$\tilde{\mathbf{X}} \sim \mathbf{P}_0^{\dagger}\mathbf{x} + \lambda\widetilde{\mathbf{C}} \quad with \quad \mathbf{P}_0^{\dagger} = \begin{bmatrix} \mathbf{K}^{-1} \\ \mathbf{0}^T \end{bmatrix} \; and \; \widetilde{\mathbf{C}} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \tag{2.17}$$

where $\tilde{\mathbf{X}}$, with abuse of the notation, represents the homogeneous coordinates in the 3D space of x, $\mathbf{P}_0^{\dagger}$ the pseudo-inverse of $\mathbf{P}_0$ and $\widetilde{\mathbf{C}}$ the homogeneous coordinates of the first camera center, noted $\mathcal{C}$. Thus, $\mathbf{l}'$ is computed as the line joining the projection into the second view of the first camera center $\mathbf{P'}_0\widetilde{\mathbf{C}}$ and the point $\tilde{\mathbf{X}}$ at the scale $\lambda = 0$,

$$\begin{aligned} \mathbf{l}' &= (\mathbf{P'}_0\widetilde{\mathbf{C}}) \times (\mathbf{P'}_0\mathbf{P}_0^{\dagger}\mathbf{x}) && (2.18)\\ &= \mathbf{F}\mathbf{x} && (2.19) \end{aligned}$$

where the fundamental matrix is algebraically computed as:

$$\mathbf{F} = \left[\mathbf{P}'_0\widetilde{\mathbf{C}}\right]_{\times} \mathbf{P}'_0\mathbf{P}_0^{\dagger} \tag{2.20}$$

Different forms of the fundamental matrix can result by replacing the corresponding variables in Eq. 2.20. Some of them are provided by the following,

$$\begin{aligned} \mathbf{F} &= [\mathbf{K}'\,\mathbf{t}]_{\times}\,\mathbf{K}'\mathbf{R}\mathbf{K}^{-1} && (2.21)\\ &= \mathbf{K}'^{-T}\,[\mathbf{t}]_{\times}\,\mathbf{R}\mathbf{K}^{-1} && (2.22) \end{aligned}$$

Eq. 2.19 not only provides an algebraic expression to compute the epipolar line $\mathbf{l}'$ but also encloses a practical property of the fundamental matrix. This property establishes that a 3D point observed in two different views must satisfies the following statement

$$\mathbf{x}'^{T}\mathbf{F}\mathbf{x} = 0 \tag{2.23}$$

The specialized representation of the fundamental matrix for the case of normalized camera matrices is the so called *essential matrix*. This representation assumes the use of calibrated cameras, hence, the effects of the camera calibration matrix are removed by considering it as the identity matrix.

From the above, consider two *normalized* camera matrices (canonical form) respectively denoted $\mathbf{P} = [\mathbb{I}\,\mathbf{0}]$ and $\mathbf{P}' = [\mathbf{R}\,\mathbf{t}]$. Thus, the corresponding essential matrix is given by

$$\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R} = \mathbf{R}[\mathbf{R}^T\,\mathbf{t}]_{\times} \tag{2.24}$$

The essential matrix preserves the fundamental matrix property, stated in Eq. 2.23,

for normalized image coordinates as follows,

$$(\mathbf{x}'^T\mathbf{K}'^{-T})\mathbf{E}(\mathbf{K}^{-1}\mathbf{x}) = 0 \tag{2.25}$$

Monocular Vision

Two-view geometry constraints can be considered for a moving monocular system over time (two samples) which is observing a static scene. So far, the estimation of the fundamental and the essential matrices have been addressed based on the prior knowledge of the camera view points (i.e extrinsic parameters). Now, we focus on the estimation of the camera pose from measured and associated points. This inverse problem is known as *structure-from-motion* [CCG06]. To deal with, it is necessary to consider the use of two-view constraints.

Structure-from-Motion Problem Consider a calibrated camera and the associated imaged positions of at least five observed points in two views. Based on this minimal parametrization, it can be recovered the relative structure of the points and the camera motion in space (so called five-point problem) [Nis04, LH06]. In practice, however, more than five-points are needed to achieving a robust structure and motion estimation. These estimations are subject to an arbitrary projective transformation (i.e. scale).

To illustrate this, let be $\mathbf{P}_0 = \mathbf{K}[\mathbb{I}_{3\times3}\,\mathbf{0}]$ and $\mathbf{P}_1 = \mathbf{K}[\mathbf{R}\,\mathbf{t}]$, the projective matrices of a moving camera in space at two sampling times. These matrices are reduced to their canonical form, postmultiplying them by an invertible 4×4 real matrix,

$$\mathbf{H} = \begin{bmatrix} \mathbf{K}^{-1} & \mathbf{0} \\ \boldsymbol{\alpha}^T & \lambda \end{bmatrix} \tag{2.26}$$

where $\boldsymbol{\alpha}$ is a 3-vector with arbitrary elements and λ is a non-zero real number. Redefining $\mathbf{P}_0 = [\mathbb{I}_{3\times3}\,\mathbf{0}]$ and $\mathbf{P}_1 = [\mathbf{R}\,\mathbf{t}]$, it can be noticed that there still remains the scale ambiguity for

$$\mathbf{H} = \begin{bmatrix} \mathbb{I}_{3\times3} & \mathbf{0} \\ \mathbf{0}^T & \lambda \end{bmatrix} \tag{2.27}$$

λ being a scale factor not equals to 1. This scale factor means, for instance, that if we try to estimate the relative camera motion parameters, here $[\mathbf{R}\,\mathbf{t}]$, for a set of observed 3D points, the transformation $[\mathbf{R}\,\lambda\mathbf{t}]$ is a possible solution.

The projective ambiguity evidenced by λ, can be easily solved through the knowledge integration of a metric constraint between the observed 3D points. Alternatively, two cameras rigidly joint can provide a straightforward 3D estimation of the scene structure

without any ambiguity. In the next, this vision system configuration is examined focusing on the ego-motion estimation problem.

Stereo Vision

Based on the same principle as human vision, it is possible to reconstruct in a snapshoot the scene structure from stereo images. This task can be accomplished, since the correspondence between the points seen from both views is known. Finding point correspondences and computing two-view constraints constitute a "chicken-egg" dilemma. Indeed, in order to compute the fundamental matrix is necessary a set of corresponding points. Conversely, the search of the point correspondence is made based on the two-view constraints. This problem is partially solved in stereo vision systems by performing an offline calibration routine. In this way, the point correspondence (so called stereo matching) is determined through a human-supervised procedure and the two-view constraints engendered in the stereo vision system are estimated.

Once the extrinsic parameters of the stereo rig are determined, the stereo point association task can be better addressed. Additional issues must be considered also such as the image lens distortions and the optical axes misalignment of the cameras which complexifies the image point matching task. To this end, a stereo image correction known as stereo image rectification should be performed.

Stereo Image Rectification The perfect alignment of the stereo images provides interesting advantages in the stereo correspondence process. These advantages not only speed up the matching process, but also grant a computational efficient application of the two-view constraints. In practice, real stereo vision systems are never perfectly aligned. Moreover, optical image distortions are also present and can cause unreliable parings.

For this reason, the acquired images must be corrected into a exact coplanar, row-aligned configuration. Such a correction is not performed on the vision system hardware, but in the pixel image coordinates mapping. The choice of a good correction mapping is usually based on the view overlapping maximization and the minimization of the undesired effects such image distortions [FTV99].

Considering the particular case of calibrated cameras, the rotation and translation between them are known. Thus, the image alignment procedure is straightforward. To this end, distortions are corrected making use of the coefficients estimated in the camera calibration routine. Then, the left and right image rectification mappings are obtained from a two rotation composition of the image planes. Applying the rectification of every pixel location can cause empty pixel regions on the final image. This problem is dealt through a bi-linear pixel interpolation of the pixels values. Finally, the effective image

regions are cropped ensuring a maximal common view image area. This procedure is summarized in Fig. 2.10.

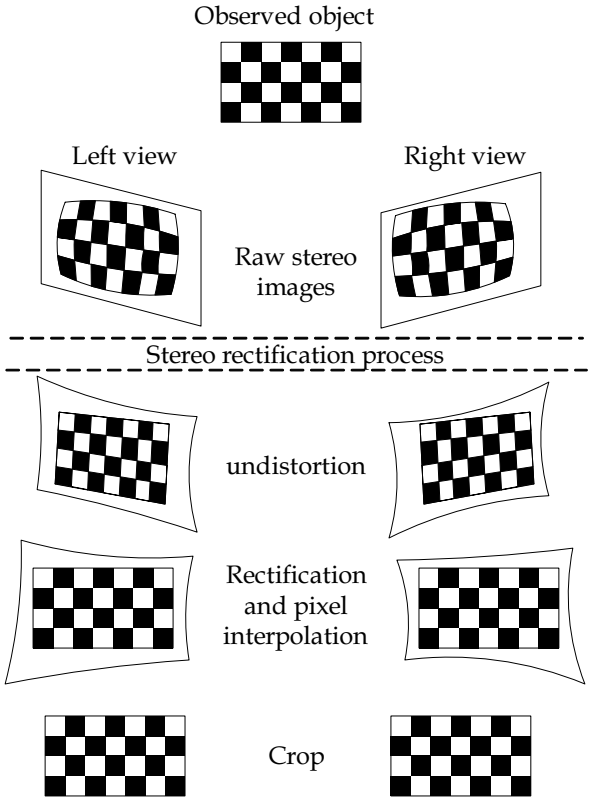


Figure 2.10: *Stereo rectification routine can be summarized in three main steps: In the undistortion step, image distortions induced by the camera lens are corrected. In the rectification step, images are "deformed" so as the image planes become coplanar and row-aligned. Then, missing pixel values are interpolated. Finally, the rectified images are cropped ensuring a good view overlapping.*

Feature-based Stereo Matching The stereo correspondence problem establishes the link between brightness and geometrical scene information. This link is crucial for achieving scene understanding and 3D structure reconstruction (i.e. objects, surfaces, etc.). Thus, image points can be associated using a discriminant enough criterion based on their appearance similarity. This criterion exploits the texture information enclosed in the images. For this, a score is computed and locally maximized by taking into account the pixel values included in a centered correlation window as illustrated in Fig. 2.11.

This technique is called feature-based matching (also known as template matching) and is well adapted for on-line applications. However, associating points by their local appearance similarity does not ensure at all the rightness of the stereo matching. For instance, template association in boundary regions of imaged objects (i.e. speckle regions) may lead to ambiguities. This is due to the background regions in the correlation window which might be occluded in one of the views. Attempting to reduce as possible all error sources, robust statistics techniques and geometrical constraints must also be

combined with the appearance criterion.

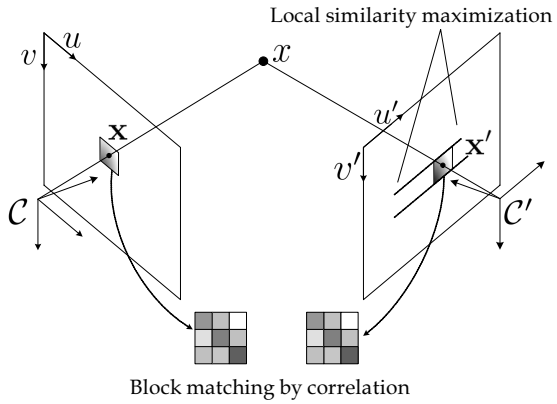


Figure 2.11: *Stereo association along the epipolar line using a block matching technique*

In the literature, one can refer to a quite large variety of similarity metric criteria as it has been presented in [SS01]. Two common similarity measurements are the Sum of Absolute Differences (SAD) and the Sum of Squared Differences (SSD). They can be easily computed as follows:

$$SAD = \frac{1}{m \times n} \sum_{i,j=1}^{m,n} |\mathbf{I}_1(i,j) - \mathbf{I}_2(i,j)| \tag{2.28}$$

$$SSD = \frac{1}{m \times n} \sum_{i,j=1}^{m,n} (\mathbf{I}_1(i,j) - \mathbf{I}_2(i,j))^2 \tag{2.29}$$

where m and n are the dimensions of the centered templates $\mathbf{I}_1$ and $\mathbf{I}_2$. The minimal value of these metrics is associated with the highest similarity of the compared templates. Based on simple brightness differences, these criteria can be implemented with a low computational cost. Nevertheless, such criteria are very sensitive to image noise. A better choice for associating photometric features is the use of a correlation measurement such as the Normalized Cross-Correlation (NCC) and the Zero-mean Normalized Cross-Correlation (ZNCC). A high discriminant effectiveness can be achieved using such correlation measurements since they are invariant to scalings and shifts of brightness intensities which are due to illumination changes over time. The NCC and the ZNCC correlations, estimated in a block-matching window of $m \times n$ pixels centered around an image point, can be computed through the following expressions:

$$NCC = \frac{\sum_{i,j=1}^{m,n} (\mathbf{I}_1(i,j))\,(\mathbf{I}_2(i,j))}{\sqrt{\sum_{i,j=1}^{m,n} (\mathbf{I}_1(i,j))^2 \sum_{i,j=1}^{m,n} (\mathbf{I}_2(i,j))^2}} \tag{2.30}$$

$$ZNCC = \frac{\sum_{i,j=1}^{M,N} \left(\mathbf{I}_1(i,j) - \bar{\mathbf{I}}_1\right)\left(\mathbf{I}_2(i,j) - \bar{\mathbf{I}}_2\right)}{\sqrt{\sum_{i,j=1}^{m,n} \left(\mathbf{I}_1(i,j) - \bar{\mathbf{I}}_1\right)^2 \sum_{i,j=1}^{m,n} \left(\mathbf{I}_2(i,j) - \bar{\mathbf{I}}_2\right)^2}} \tag{2.31}$$

where $\bar{\mathbf{I}}_1$, $\bar{\mathbf{I}}_2$ are the mean intensity values of the template images $\mathbf{I}_1$ and $\mathbf{I}_2$. As one can notice, the robustness achieved by these correlation criteria is obtained thanks to normalized brightness differences, at the expense of a higher computational complexity. The template similarity is then expressed within a normalized interval, $[-1 + 1]$, independently of the window correlation size. Thus, a correlation value of 1 indicates the highest template likeness.

Associating image points based only on their brightness similarity in a dense way, constitutes a time consuming task doomed to several erroneous matching. Attempting to improve the performance and the robustness of the feature-based stereo matching, different strategies can be additionally adopted.

An interesting approach is the sparse stereo matching, since it speeds up the correspondence process by reducing the association to a set of rich texture features (see Section 2.2) and preserves the key information of the scene structure. Stereo vision systems induce geometrical constraints from the projective principle of the image formation. The use of these two view properties reduces the number of image point candidates to be evaluated in the matching process, since the location of the corresponding image point can be constrained.

Suppose that a set of keypoints have been extracted in the left and right images. Then, the search space of the corresponding point of a left feature point can be drastically limited to the keypoints which lie in the right epipolar line. For the particular case where the provided images are rectified, the search is reduced to include only the keypoints located at the same row image line (i.e. v image coordinate) as illustrated in Fig. 2.12

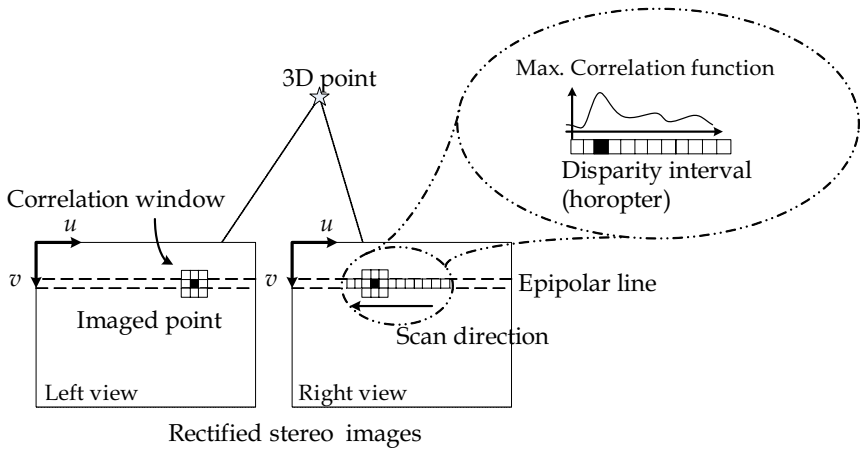


Figure 2.12: *Stereo Matching in Rectified Images*

Disparity constraint Visual odometry relies on observed key features over time for estimating the ego-motion performed by the moving vision system. This is why it is necessary to keep viewing key features as long as possible. Consider that the vision system is aimed forward in the same direction as the motion. Thus, the key features

corresponding to objects which are too close to the vision system will go out of the field of view quickly. In order to select features at an appropriate distance form the vision system, the stereo matching search can be constrained to a defined disparity interval. This constraint cuts down the stereo matched features to only those contained in a defined 3D volume attempting, in this way, to ensure their visibility over time.

The disparity, noted d in pixels, between two associated features is directly linked to the depth distance of the corresponding 3D point as follows,

$$z = \frac{fb}{d} \quad with \quad d = \mathbf{x}_{(u)} - \mathbf{x}'_{(u)} \tag{2.32}$$

where z is the coordinate in the depth axis direction, b the baseline distance in metric units, f the focal distance in pixels and, $\mathbf{x}_{(u)}$ and $\mathbf{x}'_{(u)}$ are the u-image coordinates of the associated features. Therefore, the disparity is restrained to an interval limit establishing the horopter of the stereo vision system. The horopter only depends on the baseline distance, the focal length and the disparity limit. Hereafter, the baseline and the focal length are considered invariant over time and only the disparity limits are used to induce changes in the horopter of the vision system. The disparity limits are defined in the 3D space by two planes fixed in depth from the cameras as illustrated in Fig. 2.13.

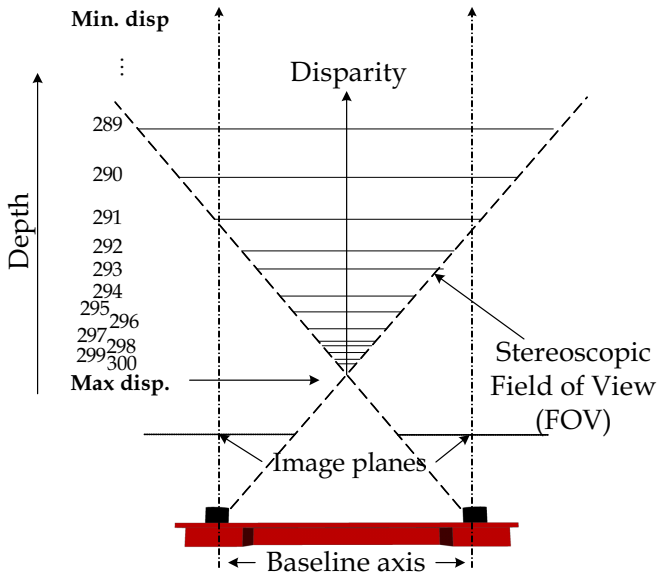


Figure 2.13: *Disparity Limits represented in the 3D Space*

For vehicle applications in particular, the horopter can be dynamically set as a function of the platform velocity. Thus, the disparity is bounded to the interval $[d_{min}\ d_{max}]$ representing, in the 3D space, the maximal and the minimal depth distance from the cameras to be considered. The following expressions associate, through a linear function, the changes induced in the disparity bounds with respect to the velocity,

$$d_{min} = \frac{bf}{z_{max}}, \quad d_{max} = \frac{bf}{z_{min}} \tag{2.33}$$

with

$$z_{max} = h_{max} + v\frac{\Delta z}{v_{max}} \quad and \quad z_{min} = h_{min} + v\frac{\Delta z}{v_{max}} \tag{2.34}$$

where the interval $[z_{min} \ z_{max}]$ denotes the depth metric bounds computed from the default depth interval in meters $[h_{min} \, h_{max}]$, v the platform velocity, v_{max} the maximal speed of the vehicle and Δz the maximal offset to be applied to the horopters. The dynamic configuration of the horopter regarding the vehicle speed is depicted in Fig. 2.14.

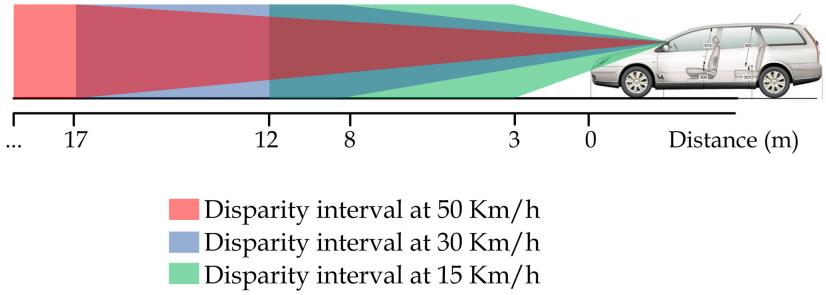


Figure 2.14: *Dynamic Horopter*

Cross-check association method The occlusion is a particular problem which occurs when a real point in space can only be visible from one of the vantage points (i.e. half-occlusion). This inevitable fact happens because of the scene geometry or the obstruction caused by other objects. As represented in Fig. 2.15, the object B is almost occluded in the right camera view, $\mathcal{C}'$, by the object A. Thus, the 3D point x_2 lying in the object B imaged at $\mathbf{x}_2$ in the left view $\mathcal{C}$, is not visible in the right one. This point cannot be associated and is hence considered as an orphan point. However, the image point $\mathbf{x}'_3$ corresponding to a 3D point lying in the object A, may lead to an incorrect association to $\mathbf{x}_2$. On the other hand, the 3D point x_1 in the object A is completely visible from both camera views. Hence, the image points $\mathbf{x}_1$ and $\mathbf{x}'_1$ can be correctly paired.

Occlusions do not make the stereo matching task easier since they can lead the algorithm to incorrect pairings. Thus, this problem is addressed using a cross-check association [CS09, NNB04, NNB06]. For this, the feature-based stereo matching process is performed in two steps which are detailed in Alg. 2.4.2. First, the left features are considered as a reference and they are associated to the right ones. Conversely in the second stage, the right features are kept as a reference and then associated to the left ones. Finally, only those features which have been associated in both steps are considered correct, otherwise the feature point is probably occluded. In practice, this

test can be efficiently implemented by checking the disparity values computed for each point at the two steps.

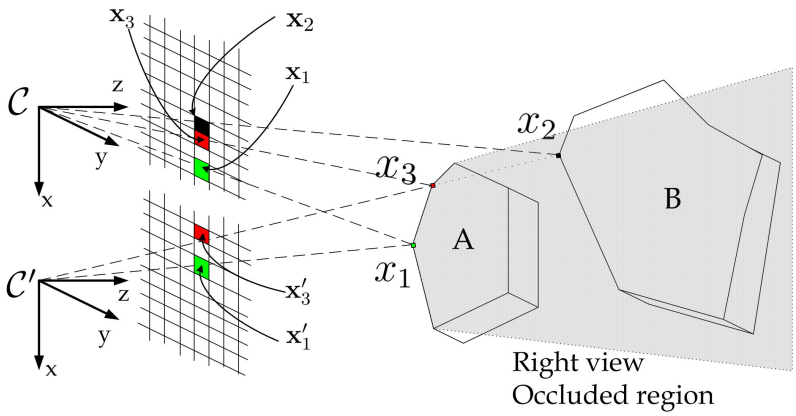


Figure 2.15: *Occlusions*

In the literature, alternative techniques to address the half-occlusion problem are also proposed. For instance, the point Ordering Constraint (ORD) consists in detecting the potential occluded points by checking the order of the associated features along the epipolar line obtained after the two-way matching (i.e. left-right, right-left).

Algorithm 2.4.2 Cross-check association algorithm

Input: - Gray scale stereo image pair or feature descriptors

- Left and right extracted keypoints coordinates $\mathsf{p}^* = \left\{p^*_{1,\ldots,i,\ldots}\right\}$ and $\mathsf{p}'^* = \{{p'}^*_{1,\ldots,j,\ldots}\}$

Output: Associated keypoints

---

for $i = 0$ **do** n ▷ First matching process
  for $j = 0$ **do** n
    ▶ Associate p^*_i located at (u, v) with respect to the right image features p'^*_j
  end for
  ▶ Update the sparse disparity map referenced to the left image, $\mathbf{D}_d(u, v)$
end for
for $i = 0$ **do** n ▷ Second matching process
  for $j = 0$ **do** n
    ▶ Associate p'^*_i located at (u', v') with respect to the left image features p^*_j
  end for
  ▶ Update the sparse disparity map referenced to the right image, $\mathbf{D}'_d(u', v')$
end for
if $\mathbf{D}_d(u, v) = \mathbf{D}'_d(u' + \mathbf{D}_d(u, v), v')$ **then**
  ▶ Correct match
else
  ▶ A feasible occluded feature point
end if

Algorithm: Feature-based stereo matching Given a rectified stereo image pair, $\mathbf{I}$ and $\mathbf{I}'$, corresponding to the left and right views of a scene and a set of key points selected for each image, p^* and p'^*, the complete feature-based stereo matching is summarized in Alg. 2.4.3. It takes into account the brightness information and the geometrical constraints induced by the stereoscopic vision system. Attempting to detect potential occlusions, a cross-check validation is performed. In real time applications, this algorithm can be easily implemented and parallelized.

Algorithm 2.4.3 Feature-based stereo matching algorithm

Input: - Gray scale stereo image pair, $\mathbf{I}$ and $\mathbf{I}'$
- Left and right extracted keypoints coordinates $\mathsf{p}^* = \left\{p^*_{1,\ldots,i,\ldots}\right\}$ and $\mathsf{p}'^* = \{{p'^*}_{1,\ldots,j,\ldots}\}$
- Template correlation size (typically 9×9 or 11×11)
- Disparity interval $\rightarrowtail [d_{min},\, d_{max}]$
- Epipolar alignment error tolerance $\rightarrowtail Th_{epipolar}$ (typically ± 2 pixels)
- ZNCC minimum acceptable value $\rightarrowtail Th_{ZNCC}$ (typically 0.7 to 0.8)

Output: Matched feature points

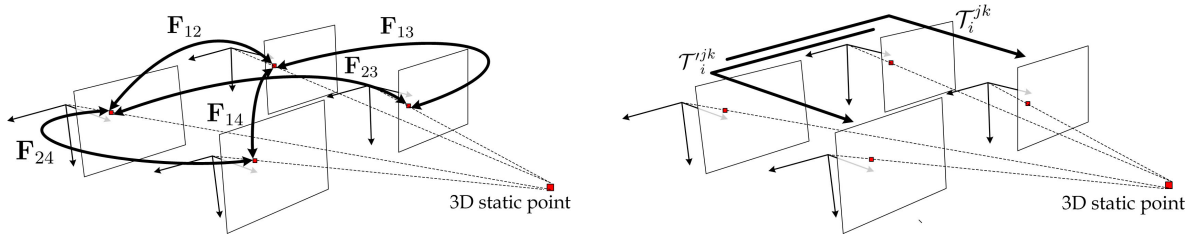
for $i = 0$ **do** n ▷ First matching process
  ▶ Extract template at $\mathbf{I}(u, v)$, around the feature point p_i^*
  for $j = 0$ **do** n
    ▶ Update disparity constraint and set $ZNCC_{last} \leftarrowtail 0$
    if $Th_{epipolar}$ and $[d_{min},\, d_{max}]$ constraints are satisfied **then**
      ▶ Extract template at $\mathbf{I}'(u', v')$, around the feature point $p_j'^*$
      ▶ Compute correlation value $ZNCC(\mathbf{I}(u, v),\, \mathbf{I}'(u,' v'))$
      if $ZNCC > Th_{ZNCC}$ and $ZNCC > ZNCC_{last}$ **then**
        ▶ Update $ZNCC_{last} \leftarrowtail ZNCC$
        ▶ Associate p_i^* to the right image feature $p_j'^*$
      end if
    end if
  end for
  ▶ Update the sparse disparity map referenced to the left image, $\mathbf{D}_d(u, v)$
end for
for $i = 0$ **do** n ▷ Second matching process
  ▶ Extract template at $\mathbf{I}'(u', v')$, around the feature point $p_i'^*$
  for $j = 0$ **do** n
    ▶ Update disparity constraint and set $ZNCC_{last} \leftarrowtail 0$
    if $Th_{epipolar}$ and $[d_{min},\, d_{max}]$ constraints are satisfied **then**
      ▶ Extract template at $\mathbf{I}(u, v)$, around the feature point ${p_j}^*$
      ▶ Compute correlation value $ZNCC(\mathbf{I}'(u', v'),\, \mathbf{I}(u, v))$
      if $ZNCC > Th_{ZNCC}$ and $ZNCC > ZNCC_{last}$ **then**

24: ▶ Update $ZNCC_{last} \leftarrow ZNCC$
25: ▶ Associate $p_i'^*$ to the left image feature p_j^*
26: **end if**
27: **end if**
28: **end for**
29: ▶ Update the sparse disparity map referenced to the right image, $\mathbf{D}'_d(u, v)$
30: **end for**
31: **if** $\mathbf{D}_d(u, v) = \mathbf{D}'_d(u' + \mathbf{D}_d(u, v), v')$ **then**
32: ▶ Correct match
33: **end if**

2.4.2 Multiple View Geometry over Time

In the previous subsection, a stereo feature tracking technique has been presented. This function "measures" the apparent motion (i.e. displacement) of the key features during two succeeding times. Hereafter, the geometric constraints engendered by multiple views (i.e. two succeeding stereo images pairs) are studied in order to "predict" the image displacement of the key features after performing a 3D rigid motion of the SVS. Suppose a static scene where a stereo point pair is obtained from of a 3D point observed at $t-1$. These points can be "transferred" to the stereo view at t, since the camera parameters and the performed motion are known.

As presented in section 2.4.1, the fundamental matrix encloses the geometric relations (i.e. projective) between two views independently of the scene structure. Indeed, multiple dependent fundamental matrices can be defined for two succeeding stereo images pairs since they are arranged by couples of vantage views as depicted in Fig. 2.16a. For more details about the fundamental matrices for multiple views and their interdependency please refer to [HZ03].



(a) *Multiple views transfer through fundamental matrices*

(b) *Multiple views transfer through trifocal tensors*

Figure 2.16: *Multiple View relations using fundamental matrices and Trifocal Tensors*

Like the fundamental matrix, the trifocal tensor encapsulates the geometrical relations between three views in a more compact representation (see Fig. 2.16b). It is composed of 27 elements which can be figured in a $3 \times 3 \times 3$ cube (or three 3×3 slides) represen-

tation. Such a cube keeps the consistency of the geometrical information transferred through the different views.

In the following, a short introduction about the theoretical basis and the geometrical information transfer through the trifocal tensor is provided. Then, the multiple view parametrization, applied to the ego-motion estimation problem is presented. For this, the Einstein tensor notation is adopted. Dealing with this notation involves some specific concepts which have been included in appendix B.

Derivation of the trifocal tensor for calibrated cameras

Consider three cameras, $\mathcal{C}$, $\mathcal{C}'$ and $\mathcal{C}''$, at different view points defined respectively by their canonical form of their projective matrices as,

$$\mathbf{P}_0 = \left[\begin{array}{cc} I_{3\times3} & \mathbf{0}_{3\times1} \end{array}\right], \mathbf{P}'_0 = \left[\begin{array}{cc} \mathbf{H}'_\infty & \mathbf{a}_4 \end{array}\right] \; and \; \mathbf{P}''_0 = \left[\begin{array}{cc} \mathbf{H}''_\infty & \mathbf{b}_4 \end{array}\right]$$

where $\mathbf{H}'_\infty$ and $\mathbf{H}''_\infty$ are 3×3 matrices representing the infinite homographies of the second and third cameras and, $\mathbf{a}_i$ and $\mathbf{b}_i$ denote the i^{th} column composing the 3×4 camera matrices. More specifically, $\mathbf{a}_4$ and $\mathbf{b}_4$ represent the epipoles of the second and the third views (i.e. $\mathbf{P}'_0$ and $\mathbf{P}''_0$) generated by the first camera center. The use of canonical cameras defines a common 3D space referenced with respect to the first camera view.

Suppose, a single line L in the 3D space imaged at $\mathbf{l}$, $\mathbf{l}'$ and $\mathbf{l}''$ in the three camera views. This single line represents the intersection in space of the three planes derived from the back-projection of the imaged line in each camera view as depicted in the Fig. 2.17.

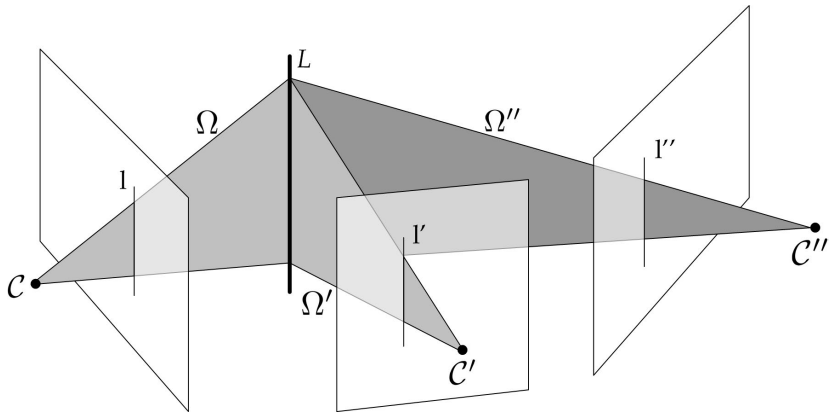


Figure 2.17: *Back-projection of the 3 lines defining 3D planes intersecting in space*

As shown in Fig. 2.17, the planes noted Ω, Ω' and Ω'' are given by:

$$\Omega = \mathbf{P}_0^T\mathbf{l} = \left[\begin{array}{c} \mathbf{l} \\ 0 \end{array}\right] \quad \Omega' = \mathbf{P}_0'^T\mathbf{l}' = \left[\begin{array}{c} \mathbf{H}_\infty'^T\mathbf{l}' \\ \mathbf{a}_4^T\mathbf{l}' \end{array}\right] \quad and \quad \Omega'' = \mathbf{P}_0''^T\mathbf{l}'' \left[\begin{array}{c} \mathbf{H}_\infty''^T\mathbf{l}'' \\ \mathbf{b}_4^T\mathbf{l}'' \end{array}\right] \tag{2.35}$$

These planes are related, since they intersect with each other at the line L. This relation can be algebraically expressed making use of the Plüker matrix representation for a line [HZ03, p. 68-73] formed by the intersection of three planes as follows [HZ03, p. 366-367]:

$$\mathbf{M} = \begin{bmatrix} \Omega & \Omega' & \Omega'' \end{bmatrix} = \begin{bmatrix} \mathbf{l} & \mathbf{H}_{\infty}'^{T}\mathbf{l}' & \mathbf{H}_{\infty}''^{T}\mathbf{l}'' \\ 0 & \mathbf{a}_4^T\mathbf{l}' & \mathbf{b}_4^T\mathbf{l}'' \end{bmatrix} \tag{2.36}$$

where $\mathbf{M}$ is a 4×3 matrix. The fact that $\mathbf{M}$ is a rank-2 matrix proves the linear dependence between its columns which can be expressed as,

$$\Omega = \alpha\,\Omega' + \beta\,\Omega'' \tag{2.37}$$

Replacing the bottom line of $\mathbf{M}$ in Eq. 2.37 yields,

$$0 = \alpha\,\left(\mathbf{a}_4^T\mathbf{l}'\right) + \beta\,\left(\mathbf{b}_4^T\mathbf{l}''\right)$$

with α and β defined as,

$$\alpha = k\,\left(\mathbf{b}_4^T\mathbf{l}''\right) \quad and \quad \beta = -k\,\left(\mathbf{a}_4^T\mathbf{l}'\right) \tag{2.38}$$

where k is a constant.
Now, applying this relation to the upper line of $\mathbf{M}$, up to a homogeneous scale factor,

$$\mathbf{l} = \left(\mathbf{b}_4^T\mathbf{l}''\right)\mathbf{H}_{\infty}'^{T}\mathbf{l}' - \left(\mathbf{a}_4^T\mathbf{l}'\right)\mathbf{H}_{\infty}''^{T}\mathbf{l}'' = \left(\mathbf{l}''^{T}\mathbf{b}_4\right)\mathbf{H}_{\infty}'^{T}\mathbf{l}' - \left(\mathbf{l}'^{T}\mathbf{a}_4\right)\mathbf{H}_{\infty}''^{T}\mathbf{l}''$$

Thus, rewriting this geometrical relation for the i^{th} element of $\mathbf{l}$, noted l_i, yields

$$l_i = \mathbf{l}''^{T}\left(\mathbf{b}_4\mathbf{a}_i^T\right)\mathbf{l}' - \mathbf{l}'^{T}\left(\mathbf{a}_4\mathbf{b}_i^T\right)\mathbf{l}'' = \mathbf{l}'^{T}\left(\mathbf{a}_i\mathbf{b}_4^T\right)\mathbf{l}'' - \mathbf{l}'^{T}\left(\mathbf{a}_4\mathbf{b}_i^T\right)\mathbf{l}''$$

simplified as,

$$l_i = \mathbf{l}'^{T}\mathbf{T}_i\mathbf{l}'' \quad with \quad \mathbf{T}_i = \mathbf{a}_i\mathbf{b}_4^T - \mathbf{a}_4\mathbf{b}_i^T \tag{2.39}$$

where i spans the 3 elements of $\mathbf{l}$. The set of matrices noted $\mathbf{T}_i$ represents the trifocal tensor. Hereafter, the trifocal tensor is dealt with in the standard tensor notation (please refer to appendix B) as follows,

$$\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k \tag{2.40}$$

Geometric transfer through multiple views

The knowledge of the intrinsic and extrinsic parameters of multiple cameras, allows the geometrical mapping of the information between their different views. Such a map-

ping does not take into account the image content but only considers the geometrical relations between views.

An elementary exchange problem between views is the point transfer. This problem is addressed in order to predict the image location of the observed keypoints after a 3D motion of the SVS. As illustrated in Fig. 2.18, it consists in determining the image position, in a third view, of a point observed in two other view points.

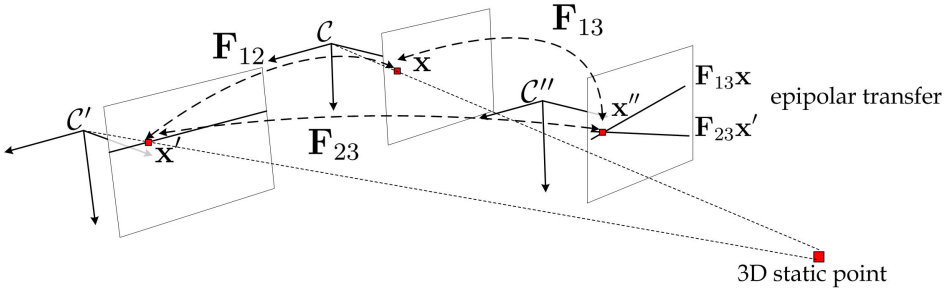


Figure 2.18: *Point Transfer between multiple views using fundamental matrices*

A first solution to the point transfer problem may be achieved using the fundamental matrix. Consider $\mathbf{x}$ and $\mathbf{x}'$, the homogeneous image coordinates of two corresponding points in the first and second view as shown in Fig. 2.18. The fundamental matrices between the cameras $\mathcal{C}$ and $\mathcal{C}''$, noted $\mathbf{F}_{13}$, and the cameras $\mathcal{C}'$ and $\mathcal{C}''$, noted $\mathbf{F}_{23}$ can be estimated since the extrinsic and intrinsic parameters are known. Thus, the epipolar line where a point $\mathbf{x}$ maps in the third view is defined by $\mathbf{F}_{13}\mathbf{x}$. In a similar way, the epipolar line where the corresponding point $\mathbf{x}'$ maps in the third view is obtained as $\mathbf{F}_{23}\mathbf{x}'$. The intersection between the epipolar lines $\mathbf{F}_{13}\mathbf{x}$ and $\mathbf{F}_{23}\mathbf{x}'$ determines the image position of the point $\mathbf{x}$ transferred into the third view as follows:

$$\mathbf{x}'' = (\mathbf{F}_{13}\mathbf{x}) \times (\mathbf{F}_{23}\mathbf{x}') \tag{2.41}$$

where $\times$ denotes the cross-product.

The point transfer solution using fundamental matrices contains a serious deficiency in a particular configuration which often dooms this method to fail. This degenerated configuration occurs when the three camera centers involved in the transfer and the imaged 3D point are close to be coplanar (i.e. close to the *trifocal plane*). Under this condition, the epipolar lines transferred to the third view become collinear. Thus, their intersection is ill-conditioned and quite inaccurate. When the coplanarity occurs the transfer is undefined.

This degenerated configuration makes the multiple view transfer through the fundamental matrix quite restrictive for practical applications. However, as will be explain later in this section, the trifocal tensor helps us avoid this issue. Thus, in a similar way as for the epipolar transfer, the homogeneous coordinates of the corresponding point in the third view, $\mathbf{x}''$, are given by the following tensorial expression:

$$x''^k = x^i l'_j \mathcal{T}_i^{jk} \tag{2.42}$$

where x''^k is the contravariant representation of $\mathbf{x}''$ and x^i is the contravariant vector representing the homogeneous coordinates of the point to be transferred $\mathbf{x}$, located in the first view. l'_j is the covariant vector representing a line passing through the point $\mathbf{x}'$ in the second view and $\mathcal{T}_i^{jk}$ is the trifocal tensor. The Fig. 2.19 illustrates the point-line-point transfer from the cameras $\mathcal{C}$ and $\mathcal{C}'$ into $\mathcal{C}''$.

It should be noted that the appropriate choice of l'_j ensures the avoidance of the point transfer degenerate configuration. Different methods have been proposed to select l'_j, but many of them might be a degeneracy overkilling. An easy-to-exploit alternative consists in choosing l'_j as the line passing through $\mathbf{x}''$, orthogonal to the epipolar line $\mathbf{l}_e = \mathbf{F}_{12}\mathbf{x}$ [HZ03]. This simple condition provides a well conditioned ray-plane intersection generated from the back-projection of l'_j and $\mathbf{x}$ in the cameras $\mathcal{C}'$ and $\mathcal{C}$ respectively. Thus, l'_j is given by:

$$l'_j = \begin{bmatrix} \mathbf{l}_{e(2)} \\ -\mathbf{l}_{e(1)} \\ -\mathbf{x}_{(u)}\mathbf{l}_{e(2)} + \mathbf{x}_{(v)}\mathbf{l}_{e(1)} \end{bmatrix} \tag{2.43}$$

where $\mathbf{l}_e = \left[\mathbf{l}_{e(1)},\, \mathbf{l}_{e(2)}\, \mathbf{l}_{e(3)}\right]^T$ and $\mathbf{x} = [u\, v\, 1]^T$.

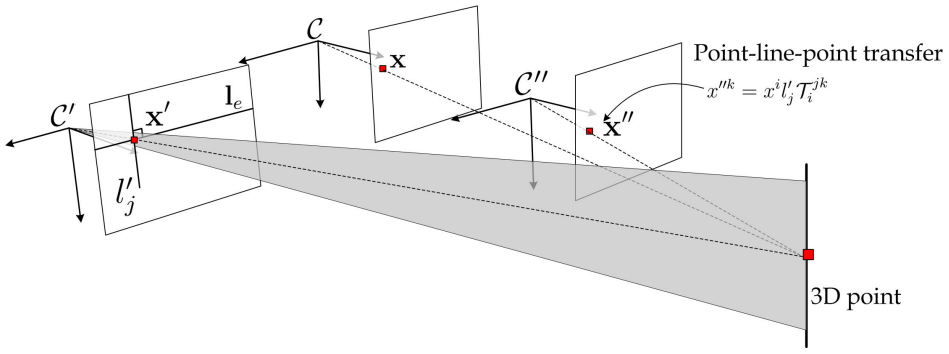


Figure 2.19: *Point transfer through the trifocal tensor*

2.5 Proposed 3D Visual Odometry Method

The proposed visual odometry algorithm starts by extracting an initial set of features from synchronized stereo images. The number of extracted features which are kept for each image was limited, bounding in this way the execution time. Let p^* and p'^* be the sets of extracted keypoints from the left and right image respectively which are acquired at time t. These sets contain the corresponding homogeneous image coordinates of the features as follows:

$$\mathsf{p}^* = \left\{[u,\, v,\, 1]_1^T, \ldots, [u,\, v,\, 1]_i^T, \ldots\right\} \tag{2.44}$$

$$\mathsf{p}'^* = \left\{[u',\, v',\, 1]_1^T, \ldots, [u',\, v',\, 1]_j^T, \ldots\right\} \tag{2.45}$$

where u, v and u', v' are sub-pixel image coordinates. i and j, are the index number of the keypoints obtained from each image.

Alternative strategies for composing the initial keypoints set have been experimented. For instance, a strategy attempting to obtain a well distributed set of keypoints, consist in partitioning the image in sectors. Then, a constant number of features by sector is extracted, ensuring scattered keypoints in all the image area. In fact, this extraction strategy helps avoiding keypoints concentrations on specific objects improving the robustness of the algorithm. However, no significant impact was observed in our experimental results.

2.5.1 Applying Feature Tracking to the Visual Odometry Problem

There are different ways to exploit feature tracking in order to estimate the 3D motion of a moving stereo vision system. In the following, a tracking scheme applied to stereoscopic images is presented. This scheme aims at accelerating and simplifying the association process of the features through frames over time. This is possible since tracking avoids continuous feature extraction from stereo image at every sample time.

The pyramidal Lucas-Kanade tracking (for short LK tracking) method allows us to measure the image position of the features at $t+1$ by minimizing the error function criterion stated in Eq. 2.7. Rewriting Eq. 2.7 in a more detailed form yields:

$$\epsilon(\mathbf{u}) = \sum_{i=u-m/2}^{u+m/2} \sum_{j=v-n/2}^{v+n/2} \left(\mathbf{I}([i + \frac{du}{dt}, j + \frac{dv}{dt}]^T, t) - \mathbf{I}([i,j]^T, t-1)\right)^2 \tag{2.46}$$

where m and n are the correlation window dimensions, $\mathbf{u}$ is the optical flow vector, $\mathbf{I}(\cdot, t-1)$ is the gray-scale image template feature to be tracked centered at (u, v) and $\mathbf{I}(\cdot, t)$ is the current gray-scale image template feature which position is estimated iteratively at different scales by the minima of the error function.

The LK tracking is performed from frame-to-frame independently in the left and the right views, without any stereo and scene rigidity constraint. Then, the tracked features in both views at t are associated using the Alg. 2.4.3 presented in section 2.4.1.

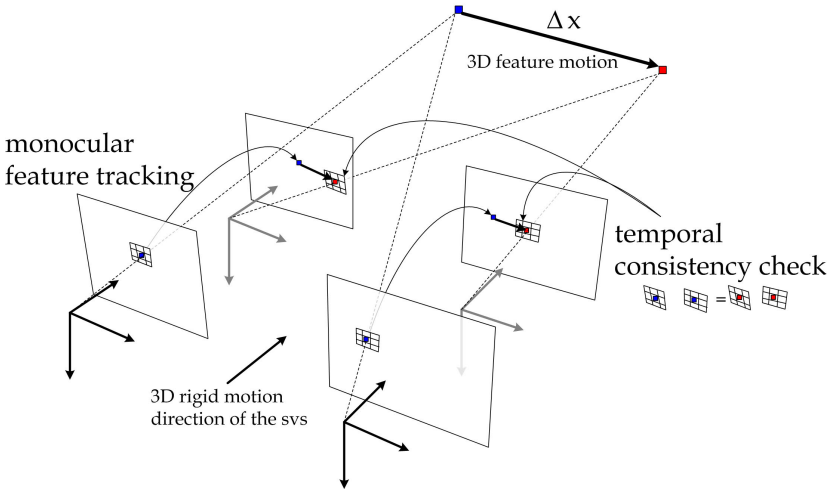


Figure 2.20: *Tracking Features in stereoscopic images*

Subsequently, the algorithm keeps only those tracked features who have been stereo associated at t in the same way as they were associated at $t-1$. This last stereo matching time invariance aims at ensuring the consistency of the observed features (i.e. consistency check). This procedure is depicted in Fig. 2.20 and summarized in the Alg. 2.1.

Algorithm 2.1 Stereo Feature Tracking

Input: - Input parameters of the feature-based stereo matching algorithm 2.4.3
- Pyramid levels (typically 3 levels for a 320×240 pixel image resolution)

Output: - Features localized in two successive stereo image pairs
- Sparse optical flow

1: ▶ Extract key features in stereo view at time $t-1$
2: ▶ Perform the feature-based stereo matching detailed in Alg. 2.4.3
3: ▶ Track features from left and right view independently
4: ▶ Perform stereo association of the tracked features at time t
5: ▶ Check tracking consistency over time regarding to the stereo feature association

2.5.2 Multiple View Parametrization for Ego-motion Estimation

In order to estimate the 3D motion of a SVS in space, the multiple view transfer formalism is extended to the four views. Four vantage points are obtained considering the moving SVS which observes a static scene at two succeeding times. Under this conditions, all the observed views must satisfy the engendered quadrilinear relations independently of the scene structure. This fact can be verified through a stereo warping operator, proposed by Comport et al. [CMR07], able to transfer with sub-pixel accuracy any corresponding point observed in a stereo image pair to a next one over time.

Consider a temporal framework composed by two rectified stereo images sampled at $t-$

1 and t, where t designates the current sampling index as depicted in Fig. 2.21. The left and right images acquired at $t-1$ are adopted as the reference views and are noted respectively $\mathbf{I}^*$ and $\mathbf{I}'^*$. In these reference frames the keypoints set are noted as $\mathsf{p}^* = \{\mathbf{p}^*_1, \ldots, \mathbf{p}^*_i, \ldots, \mathbf{p}^*_s\} \in \mathbf{I}^*$ and $\mathsf{p}'^* = \{\mathbf{p}'^*_1, \ldots, \mathbf{p}'^*_i, \ldots, \mathbf{p}'^*_s\} \in \mathbf{I}'^*$ where $\mathbf{p}^*_i$ and $\mathbf{p}'^*_i$ designate the homogeneous image coordinates of a stereo corresponding point. Accordingly, the left and right images acquired at t are referred to, as $\mathbf{I}''$ and $\mathbf{I}'''$.

The 3D motion of the SVS between two sample times is modeled by a rigid transformation (i.e. rotation-translation composition) between an orthonormal frame $\mathcal{S}(t-1)$, located in the middle of the SVS baseline, and its corresponding position at t, $\mathcal{S}(t)$. This transformation is noted in the following as ${}^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}]_{\mathcal{S}(t)}$ defined in the special Euclidean group $\mathbb{SE}(3)$.

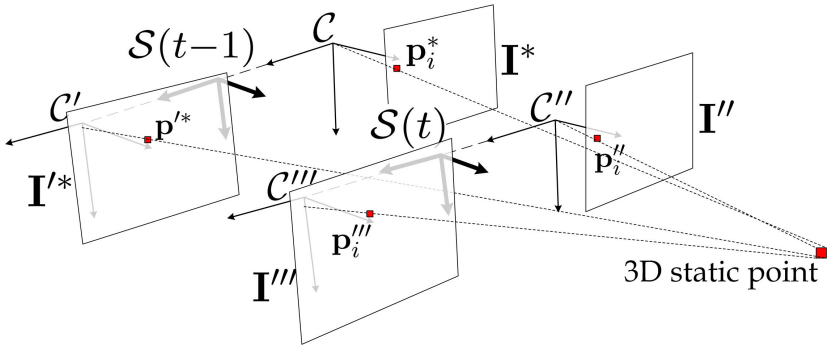


Figure 2.21: *Quadrifocal warping*

Based on the above framework description, the location of any reference stereo keypoints, $\mathbf{p}^*_i$, $\mathbf{p}'^*_i$, may be efficiently determined in the stereo image pair at time t (i.e. $\mathbf{I}''$ and $\mathbf{I}'''$), through a quadrifocal warping function

$$w({}^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}]_{\mathcal{S}(t)}\,,\ \mathbf{p}^*_i,\ \mathbf{p}'^*_i) \tag{2.47}$$

where $\Delta\boldsymbol{\omega}$ is an axis-angle rotation, $\Delta\mathbf{v}$ is a translation vector and the new estimated coordinates are denoted $\mathbf{p}''_i$ and $\mathbf{p}'''_i$.

The properties of such a warping operator, $w(\cdot)$, considerably depend on the choice of its parametrization. Different parametrization cases were evoked by Comport et al. in [CMR10]. A first possible parametrization consists in the use of one quadrifocal tensor defined, for instance, with respect to the left reference camera, $\mathcal{C}$ (see upper left section in Fig. 2.22). This configuration provides a warping operator with a low-complexity and a simple pixel accuracy. Another considered parametrization consists in defining two quadrifocal tensors, one w.r.t to the left reference camera, $\mathcal{C}$, and the other w.r.t to the right one, $\mathcal{C}'$. Contrary to the first case, the latter parametrization, depicted in the upper right section in Fig. 2.22, provides an ideal sub-pixel warping accuracy with the drawback of an important number of quadrilinear relations. This approach

remains the most adapted choice for achieving an accurate 3D motion estimation, even though it presents high computational cost.

To reduce the number of constraints provided by the two-quadrifocal approach, each tensor can be decomposed into (or constructed from) a fundamental matrix and two trifocal tensors as stated in [SW00],

$$\delta_i \mu_j \mathcal{Q}^{ijkl} = \left[\delta_i \mu_j \mathcal{T}_l^{ij}\right]_x \mathbf{F}_{34} \left[\delta_i \mu_j \mathcal{T}_k^{ij}\right]_x \tag{2.48}$$

where $\mathcal{Q}^{ijkl}$ is a quadrifocal tensor, a fourth order tensor represented by a homogeneous $3 \times 3 \times 3 \times 3$ array of elements which connects images points along four views. x is an index from the set $\{i, j, k, l\}$ and δ_i, μ_j vary to range over (1, 0, 0), (0, 1, 0), (0, 0, 1).

From Eq. 2.48, the two-quadrifocal tensor approach can be then decomposed in a total of four trifocal tensors and two fundamental matrices as depicted in lower right section of Fig 2.22.

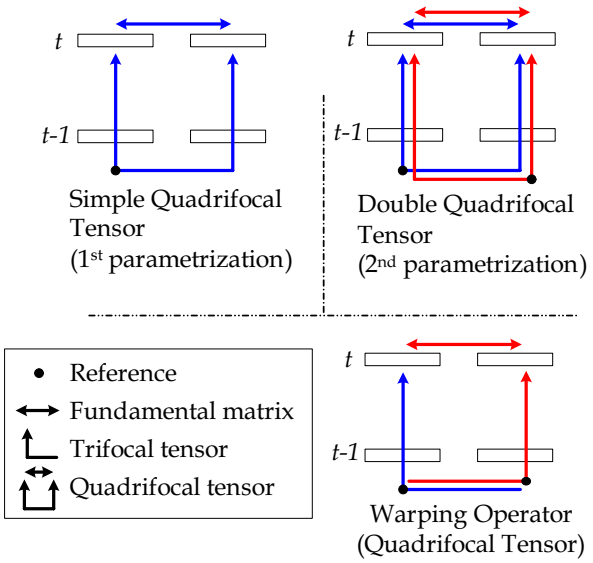


Figure 2.22: *Two-quadrifocal tensor approach decomposition*

As stated in [CMR10], the simplified constraint parametrization is constituted by the tensors relating the camera triplets $(\mathcal{C}, \mathcal{C}', \mathcal{C}'')$ and $(\mathcal{C}', \mathcal{C}, \mathcal{C}''')$, denoted respectively $\mathcal{T}_i^{jk}$, $\mathcal{T}_l^{mn}$, and the fundamental matrix $\mathbf{F}_{34}$ of $(\mathcal{C}'', \mathcal{C}''')$, which is equivalent to $\mathbf{F_{12}}$. This parametrization remains robust to the camera modeling errors and depends on the unknown motion parameters only, ${}^{\mathcal{S}(t-1)}\left[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}\right]_{\mathcal{S}(t)}$. Thus, the stereo warping operator is given by:

$$\begin{bmatrix} p''^k \\ p'''^n \end{bmatrix} = \begin{bmatrix} p^{*i} l'_j \mathcal{T}_i^{jk} \\ p'^{*l} l_m \mathcal{T}_l^{mn} \end{bmatrix} \tag{2.49}$$

where l'_j and l_m are respectively the covariant representation of the left and right image lines passing through the image points p^{*i} and p'^{*l}, and perpendicular to the epipo-

lar line. The image points p^{*i}, p'^{*l}, p''^{k}, p'''^{n} are the contravariant representations of $\mathbf{p}^*$, $\mathbf{p}'^*$, $\mathbf{p}''$, $\mathbf{p}'''$. Here, the subscript index, i, of the homogeneous coordinate representation has been omitted for clarity.

The trifocal tensors involved in the stereo warping operator (see Eq. 2.49) are functions of the intrinsic-extrinsic SVS parameters and the 3D camera motion. As the motion of the camera is referenced in a common geometrical frame $\mathcal{S}(t-1)$ (see Fig. 2.23), the formalism of canonical projective cameras is used. In the following, the computation of the trifocal tensors is detailed for a calibrated SVS.

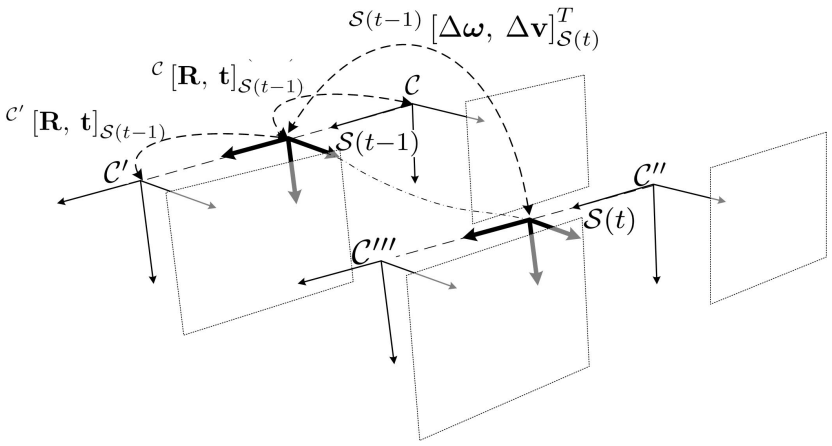


Figure 2.23: *Frame transformations and ego-motion parametrization considered for the trifocal tensors computation of the stereo warping operator*

Consider the triplet camera views ($\mathcal{C}$, $\mathcal{C}'$, $\mathcal{C}''$) associated to the tensor $\mathcal{T}_i^{jk}$. The 3×4 projective camera matrix of $\mathcal{C}$ defined in the SVS frame, $\mathcal{S}(t-1)$, is given by:

$$\mathbf{P} = \mathbf{K}[{}^{\mathcal{C}}\mathbf{R}_{\mathcal{S}(t-1)}\ {}^{\mathcal{C}}\mathbf{t}_{\mathcal{S}(t-1)}] \tag{2.50}$$

where $\mathbf{K}$ represents the left intrinsic camera parameters, ${}^{\mathcal{C}}\mathbf{R}_{\mathcal{S}(t-1)}$ is a 3×3 rotation matrix and ${}^{\mathcal{C}}\mathbf{t}_{\mathcal{S}(t-1)}$ is a 3×1 translation vector expressing respectively the orientation and the position of $\mathcal{S}(t-1)$ in $\mathcal{C}$. The rigid transformation from $\mathcal{S}(t-1)$ to $\mathcal{C}$ is denoted ${}^{\mathcal{C}}[\mathbf{R},\ \mathbf{t}]_{\mathcal{S}(t-1)}$.

To obtain the desired canonical matrix of the camera $\mathcal{C}$, noted ${}^{\mathcal{C}}\mathbf{P}_0$, let $\mathbf{U}$ be an augmented 4×4 non-singular matrix,

$$\mathbf{U} = \begin{bmatrix} \mathbf{P} \\ \mathbf{0}_{1\times 3}\ 1 \end{bmatrix}^{-1}_{4\times 4} \tag{2.51}$$

which verifies,

$${}^{\mathcal{C}}\mathbf{P}_0 = \mathbf{P}\mathbf{U} = [I_{3\times 3}\ \mathbf{0}_{3\times 1}] \tag{2.52}$$

Hence, the corresponding canonical matrices of the cameras $\mathcal{C}'$ and $\mathcal{C}''$ are:

$$
\begin{aligned}
{}^{\mathcal{C}}\mathbf{P}'_0 &= \mathbf{P}'\mathbf{U} \quad with \quad \mathbf{P}' = \mathbf{K}'[{}^{\mathcal{C}'}\mathbf{R}_{\mathcal{S}(t-1)}\ {}^{\mathcal{C}'}\mathbf{t}_{\mathcal{S}(t-1)}] && (2.53)\\
{}^{\mathcal{C}}\mathbf{P}''_0 &= \mathbf{P}''\mathbf{U} \quad with \quad \mathbf{P}'' = \mathbf{K}[{}^{\mathcal{C}''}\mathbf{R}_{\mathcal{S}(t-1)}\ {}^{\mathcal{C}''}\mathbf{t}_{\mathcal{S}(t-1)}] && (2.54)
\end{aligned}
$$

where $\mathbf{K}'$ denotes the intrinsic parameters of the right camera, ${}^{\mathcal{C}'}[\mathbf{R},\ \mathbf{t}]_{\mathcal{S}(t-1)}$ the transformation from $\mathcal{C}'$ to $\mathcal{S}(t-1)$ and ${}^{\mathcal{C}''}[\mathbf{R},\ \mathbf{t}]_{\mathcal{S}(t-1)}$ the transformation from $\mathcal{C}''$ to $\mathcal{S}(t-1)$. Recalling that the notation in ${}^{\mathcal{C}}\mathbf{P}'_0$ indicates the canonical camera matrix of $\mathcal{C}'$ with respect to the camera $\mathcal{C}$. Accordingly, ${}^{\mathcal{C}}\mathbf{P}''_0$ is the canonical matrix of $\mathcal{C}''$ w.r.t. $\mathcal{C}$. In Eq. 2.54, the intrinsic matrix of the left camera is used, since the camera parameters are considered constant over time.

Now, given the triplet camera views ($\mathcal{C}'$, $\mathcal{C}$, $\mathcal{C}'''$) and following in a similar manner the procedure for the first triplet of cameras, the canonical matrices are computed keeping as a reference the camera $\mathcal{C}'$.

Thus,

$$
\begin{aligned}
{}^{\mathcal{C}'}\mathbf{P}'_0 &= \mathbf{P}'\mathbf{V} = [I_{3\times3}\,\mathbf{0}_{3\times1}] && (2.55)\\
{}^{\mathcal{C}'}\mathbf{P}_0 &= \mathbf{P}\mathbf{V} && (2.56)\\
{}^{\mathcal{C}'}\mathbf{P}'''_0 &= \mathbf{P}'''\mathbf{V} \quad with \quad \mathbf{P}''' = \mathbf{K}'[{}^{\mathcal{C}'''}\mathbf{R}_{\mathcal{S}(t-1)}\ {}^{\mathcal{C}'''}\mathbf{t}_{\mathcal{S}(t-1)}] && (2.57)
\end{aligned}
$$

where $\mathbf{V}$ is given by,

$$
\mathbf{V} = \begin{bmatrix} \mathbf{P}' \\ \mathbf{0}_{1\times3}\ 1 \end{bmatrix}^{-1}_{4\times4} \tag{2.58}
$$

From Eq. 2.54 and Eq. 2.57, one can notice that the rigid transformations ${}^{\mathcal{C}''}[\mathbf{R},\ \mathbf{t}]_{\mathcal{S}(t-1)}$ and ${}^{\mathcal{C}'''}[\mathbf{R},\ \mathbf{t}]_{\mathcal{S}(t-1)}$ may be decomposed as functions of the 3D motion parameters of the SVS as follows:

$$
\begin{aligned}
{}^{\mathcal{C}''}\mathbf{R}_{\mathcal{S}(t-1)} &= {}^{\mathcal{C}''}\mathbf{R}_{\mathcal{S}(t)}\ {}^{\mathcal{S}(t)}\mathbf{R}_{\mathcal{S}(t-1)} && (2.59)\\
{}^{\mathcal{C}''}\mathbf{t}_{\mathcal{S}(t-1)} &= {}^{\mathcal{C}''}\mathbf{R}_{\mathcal{S}(t)}\ {}^{\mathcal{S}(t)}\mathbf{t}_{\mathcal{S}(t-1)} + {}^{\mathcal{C}''}\mathbf{t}_{\mathcal{S}(t)} && (2.60)
\end{aligned}
$$

and

$$
\begin{aligned}
{}^{\mathcal{C}'''}\mathbf{R}_{\mathcal{S}(t-1)} &= {}^{\mathcal{C}'''}\mathbf{R}_{\mathcal{S}(t)}\ {}^{\mathcal{S}(t)}\mathbf{R}_{\mathcal{S}(t-1)} && (2.61)\\
{}^{\mathcal{C}'''}\mathbf{t}_{\mathcal{S}(t-1)} &= {}^{\mathcal{C}'''}\mathbf{R}_{\mathcal{S}(t)}\ {}^{\mathcal{S}(t)}\mathbf{t}_{\mathcal{S}(t-1)} + {}^{\mathcal{C}'''}\mathbf{t}_{\mathcal{S}(t)} && (2.62)
\end{aligned}
$$

where the only non-constant transformation, ${}^{\mathcal{S}(t)}[\mathbf{R},\ \mathbf{t}]_{\mathcal{S}(t-1)}$, corresponds to the SVS ego-motion. All other frame transformations are obtained from the extrinsic calibration parameters. The defined input motion parameters of the stereo warping operator, $w(\cdot)$, corresponds to the inverse transformation induced by the ego-motion,

$$
\begin{aligned}
{}^{\mathcal{S}(t)}\mathbf{R}_{\mathcal{S}(t-1)} &= {}^{\mathcal{S}(t-1)}\mathbf{R}(\Delta\boldsymbol{\omega})^{T}_{\mathcal{S}(t)} && (2.63)\\
{}^{\mathcal{S}(t)}\mathbf{t}_{\mathcal{S}(t-1)} &= {}^{\mathcal{S}(t-1)}\mathbf{R}(\Delta\boldsymbol{\omega})^{T}_{\mathcal{S}(t)}(-\Delta\mathbf{v}) && (2.64)
\end{aligned}
$$

where $\mathbf{R}(\Delta\boldsymbol{\omega})$ means the rotation matrix corresponding to the axis-angle vector $\Delta\boldsymbol{\omega}$. Once the canonical cameras are estimated, the trifocal tensors can be computed through the use of Eq. 2.40.

2.5.3 Ego-motion Estimation

The proposed visual odometry method concludes by the formal development of a 3D motion estimation loop. Such an estimation loop is based on the feature tracking and the stereo warping operator presented in former subsections. The feature tracking represents a brightness-based technique providing the image motion measurement (i.e. *optical flow*) of a set of stereo features over time. The stereo warping operator, based on multiple view geometrical constraints, models the motion of the features in the image plane as a function of the SVS 3D ego-motion.

Assume that a static 3D point is observed from a moving SVS over two sampling times under ideal conditions (no image noise and no stereo association errors) and is tracked over time. The feature image location provided by the feature tracking at time t must match the corresponding feature image location predicted by the stereo warping operator. This condition closes the estimation loop of the 3D ego-motion.

As the major objective is to minimize the location error between the feature point location measured by the tracking and the one predicted by the warping operator, the problem can be resolved as an error optimization problem.

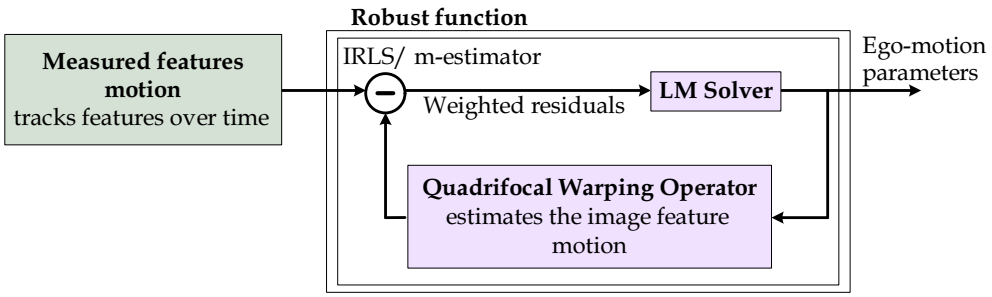


Figure 2.24: *Estimation loop of the proposed 3D visual odometry method*

We propose to estimate the 6 degrees of freedom the 3D trajectory followed by the

stereo vision system. This is done by minimizing a non-linear objective function which represents the error between the measured image motion, obtained by the tracked features, and the estimated one, obtained by the warped features.

$$f(\Delta\boldsymbol{\omega}, \Delta\mathbf{v}) = \sum_{i=1}^{s} [||\widetilde{\mathbf{p}}_i'' - \mathbf{p}_i''|| + ||\widetilde{\mathbf{p}}_i''' - \mathbf{p}_i'''||] \tag{2.65}$$

where $\widetilde{\mathbf{p}}_i''$ and $\widetilde{\mathbf{p}}_i'''$ are respectively the homogeneous coordinates of the i^{th} tracked feature points at time t. Correspondingly, $\mathbf{p}_i''$ and $\mathbf{p}_i'''$ are the homogeneous coordinates of the feature points warped by the estimated motion (see Eq. 2.49). s is the total number of tracked feature points.

Since Eq. 2.65 is a non-linear objective function, an iterative method of gradient descent is used to solve it. The Levenberg-Marquard Algorithm (LM) [Mar63, Lev44] is a standard nonlinear least-square routine able to find the global minimum by performing a downhill search. This search is guided by the Jacobian matrix of the objective function. In this way, the 3D ego-motion parameters are iteratively updated, as the residual error is minimized, until a tolerance threshold or a maximum cycle limit is reached.

The 3D visual odometry scheme holds in the rigid scene assumption. When this assumption is not satisfied, the estimation loop is broken, hence the ego-motion parameters are biased or completely incorrect. However, such an assumption is not realistic for intelligent vehicle applications, since the SVS will be immersed in environments characterized by mobile objects, complex backgrounds and occlusions. The relaxation of the rigid scene assumption can be made through the use of a robust estimator which also helps us cope with other error sources like image noise, stereo matching and tracking drifts. It should be noticed that such errors lead to breaking the same assumption.

The robust estimator mitigates the influence of outliers (i.e. feature points) which break, partially or completely, the rigidity assumption in the 3D ego-motion estimation process. The solution suggested by Hubber [Hub81] entails the use of the Iteratively Re-weighted Least Squares (IRLS) algorithm [Ste99]. The IRLS finds the maximum likelihood motion parameters by solving iteratively the following robust objective function,

$$f_{robust}(\Delta\boldsymbol{\omega}, \Delta\mathbf{v}) = \sum_{i=1}^{s} \mathbf{W}\,[||\widetilde{\mathbf{p}}_i'' - \mathbf{p}_i''|| + ||\widetilde{\mathbf{p}}_i''' - \mathbf{p}_i'''||] \tag{2.66}$$

where $f(\cdot)$ grows sub-quadratically and is monotonically non-decreasing with increasing $|r_i/\sigma_i|$ (i.e. normalized residuals absolute value) [Hub81]. In addition, σ_i^2 is the variance associated with the LM residual values, r_i, of each feature point and $\mathbf{W}$ is a weighting matrix computed through a robust M-estimator. For this, at each iterative step of the IRLS routine, Eq. 2.66 is solved using the LM algorithm with an initial

motion guess to compute new optimized estimates with the updated weighting matrix. The initialization parameters may be obtained in different ways. For instance, a naive initialization method consists in using the ego-motion estimates obtained at $t-1$ to initialize the ego-motion optimization at t. Such a method implies a constant vehicle speed assumption.

On the other hand, the updating process of the diagonal elements of $\mathbf{W}$ is done through a weight function $\rho(\cdot)$,

$$\rho(r_i/\sigma_i) = \frac{\lambda(r_i/\sigma_i)}{r_i/\sigma_i} \tag{2.67}$$

where $\lambda(\cdot)$ is an influence function modeling the outlier rejection tendency.

Different influence functions have been proposed [BT74, Hub81], where those with a rapid convergence to zero are known as "hard re-descenders" (e.g. Beaton and Tukey influence function). This kind of influence functions performs an aggressive outlier rejection and is well suited for computer vision problems [Ste99]. Hence, the Beaton and Tukey influence function is used in the solution of our motion estimation problem.

The robust routine is summarized in Alg. 2.5.4.

Algorithm 2.5.4 Iterative Re-weighted Least Squares Algorithm

---

Input: - Keypoints localized in two successive stereo image pairs based on image feature tracking (at least 30 tracked keypoints are typically used)
- A prior motion estimate ${}^{\mathcal{S}(t-1)}\,[\Delta\boldsymbol{\omega}_0,\ \Delta\mathbf{v}_0]_{\mathcal{S}(t)} \in \mathbb{SE}(3)$
- Intrinsic and extrinsic parameters of the stereo vision system
- IRLS convergence criterion $\rightarrowtail Th_{IRLS}$
- Levenberg-Marquard convergence criterion

Output: 6 DOF Ego-motion parameters ${}^{\mathcal{S}(t-1)}\,[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}]_{\mathcal{S}(t)}$

1: ▶ Initialize the LM algorithm with a prior estimate $\in \mathbb{SE}(3)$
2: ▶ Initialize weights (e.g. typically $\mathbf{W} = \mathbb{I}$)
3: **repeat**
4: ▶ Solve

$$\underset{(\Delta\boldsymbol{\omega},\Delta\mathbf{v})}{\operatorname{argmin}} \left(\sum \mathbf{W}\, [||\widetilde{\mathbf{p}}_i'' - \mathbf{p}_i''|| + ||\widetilde{\mathbf{p}}_i''' - \mathbf{p}_i'''||] \right)$$

5: ▶ Extract LM optimization residuals
6: ▶ Compute the variance associated with the LM residual values, σ_i^2
7: ▶ Update weights based on the influence function $\lambda(\cdot)$
8: ▶ Evaluate changes of the optimized motion parameters
9: **until** satisfy a given criterion Th_{IRLS}

Finally, the convergence of the IRLS optimization process is determined based on three different criteria:

– The optimized parameters changes: As the M-estimator is found, the weighing matrix values and the optimized parameters do not evolve anymore. Such changes are evaluated by thresholding.

– The maximum number of iteration cycles: Since the visual odometry function is destined for real-time applications, the convergence time should be constrained by a maximum iteration limit.

– The outlier breakdown point: This is the minimum fraction of outlying data which may lead to the divergence of an estimate. Its theoretical value corresponds to 0.5 because in presence of more than 50% of outliers, the aberrant data might minimize the estimator objective function. This criterion for the proposed application implies that more than 50% of the feature points lie on moving objects or are issued of matching and tracking errors.

2.6 Multi-modal 3D Odometry

In section 2.5, a vision-based only strategy able to provide, in real-time, positioning estimates has been presented. This method relies principally on a robust optimization process of the ego-motion parameters, ${}^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}]_{\mathcal{S}(t)}$. Being an iterative process, the optimization needs an initial guess of the motion parameters (see Alg. 2.5.4). In section 2.5.3, a naive but pragmatical initialization method based on a constant speed assumption was proposed. A more complex solution could consist in the implementation of a predictive filter of the ego-motion parameters. As the initial guess gets closer to the optimal solution, less iterations are needed to converge and the algorithm performs faster.

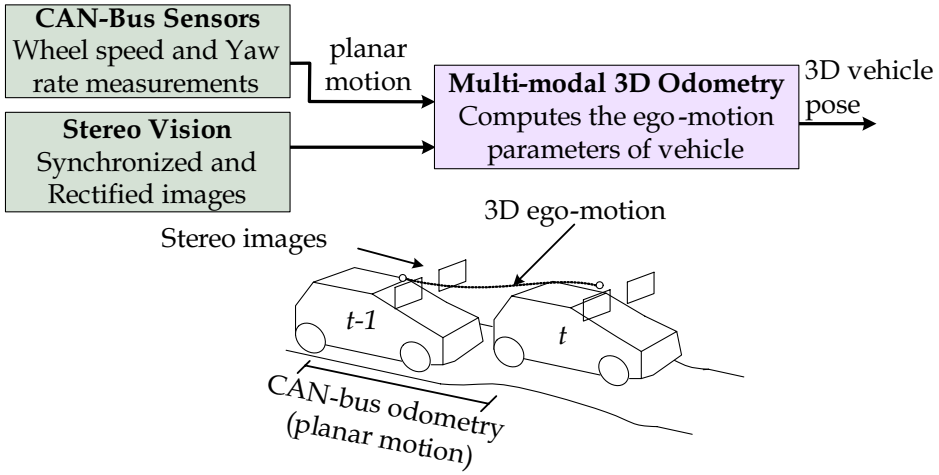


Figure 2.25: *Multi-modal 3D ego-localization scheme*

Modern vehicles are equipped with low-cost wheel speed and inertial sensors (WSS and

gyroscope) which can provide a good planar estimation of the vehicle motion. Such estimations are, however, quite sensitive to wheel slippage and road conditions.

Hereafter, a cooperative strategy implemented in a tightly coupled scheme between vision and proprioceptive sensors is detailed. This strategy consists in providing an initial planar motion guess computed using the proprioceptive sensors in an interval of time Δt. Subsequently, the 3D visual motion estimation algorithm is initialized with this motion guess (i.e. trust convergence region) and then iteratively refined (see Fig. 2.25).

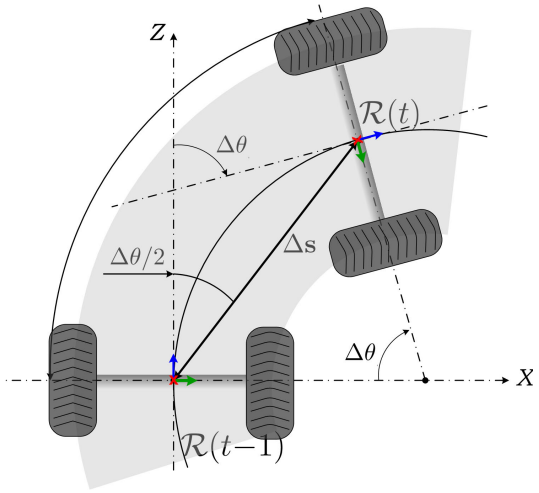


Figure 2.26: *Planar computation of the vehicle trajectory*

Let $\mathcal{R}(t)$ be the center of the body frame defined at time $t-1$. If the sampling frequency of the gyro and the WSS is high enough (about 40 Hz), the wheel speed can be considered constant during Δt, and the planar ego-motion can be approximated by a circle arc. As illustrated in Fig. 2.26, the planar ego-motion of the vehicle is modeled as follows [DJ08]:

$$\Delta\boldsymbol{\omega}_0 = \begin{bmatrix} 0 \\ \Delta\theta \\ 0 \end{bmatrix} \quad \Delta\mathbf{v}_0 = \begin{bmatrix} \Delta s \cdot \sin(\Delta\theta/2) \\ 0 \\ \Delta s \cdot \cos(\Delta\theta/2) \end{bmatrix}$$

where $\Delta\theta$ is the angle obtained by integrating the yaw rate and Δs is the integrated rear-wheel odometry in meters. $\Delta\boldsymbol{\omega}_0$ is a vector representing the axis-angle rotation of the vehicle motion and $\Delta\mathbf{v}_0$ is a vector representing the estimated displacement of the rear-wheel axis center.

The estimated motion ${}^{\mathcal{R}(t-1)}\left[\Delta\boldsymbol{\omega}_0,\ \Delta\mathbf{v}_0\right]_{\mathcal{R}(t)}$ is then considered as a near estimate of ${}^{\mathcal{S}(t-1)}\left[\Delta\boldsymbol{\omega}_0,\ \Delta\mathbf{v}_0\right]_{\mathcal{S}(t)}$. This planar motion approximation is employed in the first step of the IRLS routine (please refer to Alg. 2.5.4, line 1).

2.7 Localization using 3D Visual Odometry

One of the most classical methods employed to localize a robotic structure, consists in the incremental integration of elementary displacements. This technique is known as *dead-reckoning* or *deductive-reckoning*. In the context of intelligent vehicles, the relative vehicle localization is obtained following this classical odometry technique as illustrated in Fig.2.27. In the sequel, the local position with respect to its initial fix is formalized and an easy-to-use technique for the Geo-localization of the 3D vehicle trajectory is detailed.

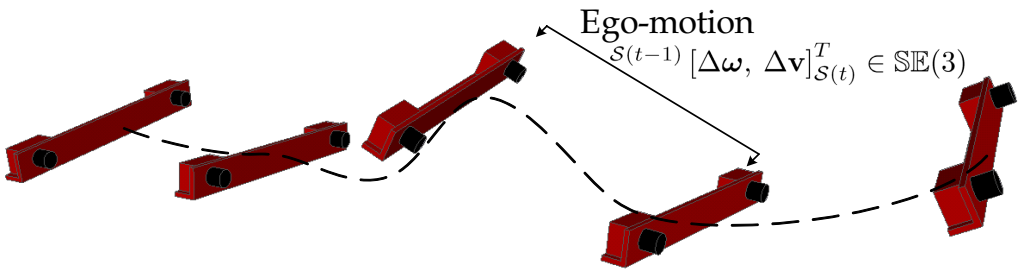


Figure 2.27: *The local vehicle position (i.e. position and attitude) is determined by the incremental use of the 3D visual odometry*

2.7.1 Odometry Integration

As odometry relies on an incremental computation of the elementary displacements, let $\mathcal{S}(t=0)$ be the initial frame position of the SVS, rigidly linked to the vehicle. Every consecutive vehicle positioning estimate is referenced with respect to this frame. Thus, the vehicle fix and attitude w.r.t. $\mathcal{S}(t=0)$ are denoted $^{\mathcal{S}}[\mathbf{q}(t),\ \mathbf{p}(t)]$which corresponds to the frame transformation, $^{\mathcal{S}(t=0)}[\mathbf{q},\ \mathbf{t}]_{\mathcal{S}(t)}$, from $\mathcal{S}(t)$ to $\mathcal{S}(t=0)$ as follows,

$$^{\mathcal{S}}\mathbf{q}(t) = {}^{\mathcal{S}(t=0)}\mathbf{q}_{\mathcal{S}(t)} \tag{2.68}$$

$$^{\mathcal{S}}\underline{\mathbf{p}}(t) = {}^{\mathcal{S}(t=0)}\underline{\mathbf{t}}_{\mathcal{S}(t)} \tag{2.69}$$

where the vehicle attitude is represented by a unit quaternion $^{\mathcal{S}}\mathbf{q}(t)$ and its 3D position by a 3×1 vector, $^{\mathcal{S}}\mathbf{p}(t)$. Underlined vectors (e.g. $\underline{\mathbf{p}}$) denote expanded forms (i.e. $\underline{\mathbf{p}} = [0,\ \mathbf{p}]^T$) for the use of the quaternion multiplication.

The positioning parameters stated in Eq. 2.68 and Eq. 2.69 are updated by concatenating the new partial motion estimates, $^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}]_{\mathcal{S}(t)}$, during the interval of time Δt as follows,

$$
\begin{aligned}
{}^{\mathcal{S}(t=0)}\mathbf{q}_{\mathcal{S}(t)} &= {}^{\mathcal{S}(t=0)}\mathbf{q}_{\mathcal{S}(t-1)} \star^{\mathcal{S}(t-1)} \mathbf{q}(\Delta\boldsymbol{\omega})_{\mathcal{S}(t)} && (2.70)\\
{}^{\mathcal{S}(t=0)}\underline{\mathbf{t}}_{\mathcal{S}(t)} &= {}^{\mathcal{S}(t=0)}\mathbf{q}_{\mathcal{S}(t-1)} \star^{\mathcal{S}(t-1)} \underline{\boldsymbol{\Delta}\mathbf{v}}_{\mathcal{S}(t)} \star^{\mathcal{S}(t=0)} \bar{\mathbf{q}}_{\mathcal{S}(-1)} +^{\mathcal{S}(t=0)} \underline{\mathbf{t}}_{\mathcal{S}(t-1)} && (2.71)
\end{aligned}
$$

where $\mathbf{q}(\Delta\boldsymbol{\omega})$ is the unit quaternion corresponding to the estimated rotation of the vehicle ego-motion ${}^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}]_{\mathcal{S}(t)}$, $\star$ denotes the multiplication quaternion operator and $\bar{\mathbf{q}}$ represents the corresponding quaternion conjugate. More details about rotation representations, compositions and transformations are provided in the appendix A.

2.7.2 Geo-localizing 3D Visual Odometry

Cooperative approaches for ADAS are based on the perception exchange between multiple vehicles [SAF]. Such approaches aim at enlarging the vehicle field of view by fusing information provided by other vehicles. For this, it is necessary to reference the local vehicle perception with respect to a global frame. GPS provides an-easy-to-exploit global frame, and allows a global localization of the 3D trajectories obtained through visual odometry. In this study, this Geo-localization is also useful to compare different localization methods.

Geo-localization consists in two principal steps: Initialization, where the transformation between the local and the Geo reference is estimated and trajectory mapping, which systematically transforms the local vehicle estimated position into Geo-referenced coordinates.

Initialization

In the initialization step, the attitude misalignment between the GPS and the visual odometry measurements is estimated. For this, a set of GPS measurements (typically 50 positioning points) is acquired simultaneously while the vehicle position is estimated through visual odometry in a local frame. For each 3D estimated vehicle position, the associated GPS fix is matched. Please notice that a GPS receiver only provides 3D positions. In order to initialize the 3D attitude, the system waits until the vehicle has traveled a given distance. Once the initialization sets of GPS and visual odometry measurements are acquired, the data misalignment can be solved as an absolute orientation problem [AHB87]. For this, the two sets of measurements must be defined in a pertinent Cartesian frame. Since the GPS measurements are often provided in geodetic coordinates in a World Geodetic System (i.e.WGS-84: latitude, longitude and ellipsoidal height), they can be converted to a Local Tangent Plane (LTP) frame, such as the East-North-Up (ENU) frame depicted in Fig. 2.28.

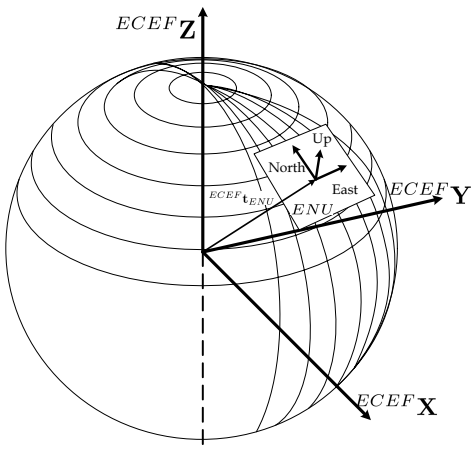


Figure 2.28: *East-North-Up Local Tangent Plane. Such a frame is usually employed in projection coordinates like Lambert 93 in France*

The WGS-84 to ENU coordinate transformation needs an intermediate conversion through the Earth-Centered-Earth-Fixed (ECEF) frame as stated in the following,

$$
\begin{aligned}
{}^{ECEF}\mathbf{p} &= llh2ecef({}^{LLH}\mathbf{p}) && (2.72)\\
{}^{ENU}\underline{\mathbf{p}} &= {}^{ENU}\mathbf{q}_{ECEF} \star^{ECEF} \underline{\mathbf{p}} \star^{ENU} \bar{\mathbf{q}}_{ECEF} +^{ENU} \underline{\mathbf{t}}_{ECEF} && (2.73)
\end{aligned}
$$

where ${}^{LLH}\mathbf{p} = [\mathbf{p}_{(lat)}, \mathbf{p}_{(lon)}, \mathbf{p}_{(alt)}]^T$ denotes the GPS measurement composed of the geodetic latitude, longitude and ellipsoidal height, ${}^{ECEF}\mathbf{p}$ is the corresponding GPS coordinates in the ECEF frame and $llh2ecef(\cdot)$ is the function performing the geodetic WGS-84 conversion into Cartesian ECEF. The final GPS point coordinates associated with the visual odometry fix estimation are represented by ${}^{ENU}\mathbf{p}$.

The ECEF-ENU rigid transformation, ${}^{ENU}[\mathbf{q}, \mathbf{t}]_{ECEF}$, is obtained as follows,

$$
\begin{aligned}
{}^{ENU}\mathbf{q}_{ECEF} &= \mathbf{q}({}^{ENU}\mathbf{R}_{ECEF}) && (2.74)\\
{}^{ENU}\underline{\mathbf{t}}_{ECEF} &= -\left({}^{ENU}\mathbf{q}_{ECEF} \star^{ECEF} \underline{\mathbf{t}}_{ENU} \star^{ENU} \bar{\mathbf{q}}_{ECEF}\right) && (2.75)
\end{aligned}
$$

with

$$
{}^{ENU}\mathbf{R}_{ECEF} = \begin{bmatrix}
-\mathrm{s}({}^{LLH}\mathbf{p}_{(lon)}) & \mathrm{c}({}^{LLH}\mathbf{p}_{(lon)}) & 0 \\
-\mathrm{c}({}^{LLH}\mathbf{p}_{(lon)}) \cdot \mathrm{s}({}^{LLH}\mathbf{p}_{(lat)}) & -\mathrm{s}({}^{LLH}\mathbf{p}_{(lon)}) \cdot \mathrm{s}({}^{LLH}\mathbf{p}_{(lat)}) & \mathrm{c}({}^{LLH}\mathbf{p}_{(lat)}) \\
\mathrm{c}({}^{LLH}\mathbf{p}_{(lon)}) \cdot \mathrm{c}({}^{LLH}\mathbf{p}_{(lat)}) & \mathrm{s}({}^{LLH}\mathbf{p}_{(lon)}) \cdot \mathrm{c}({}^{LLH}\mathbf{p}_{(lat)}) & \mathrm{s}({}^{LLH}\mathbf{p}_{(lat)})
\end{bmatrix} \quad (2.76)
$$

where $\mathbf{q}({}^{ENU}\mathbf{R}_{ECEF})$, in Eq. 2.74, is the quaternion corresponding to the rotation matrix ${}^{ENU}\mathbf{R}_{ECEF}$ and $\mathrm{c}(\cdot)$, $\mathrm{s}(\cdot)$ are respectively the cosine and sine trigonometrical

functions. It is important to highlight that ${}^{ECEF}\mathbf{t}_{ENU}$ is the first acquired GPS fix in ECEF coordinates, it defines the position of the ENU origin.

The last part of the initialization step consists in determining the attitude misalignment (i.e. rotation) between the ENU and the visual odometry space defined by the frame $\mathcal{S}(t=0)$. Only an attitude misalignment is considered here because the GPS antenna is placed at the SVS frame origin (i.e. midpoint of the SVS). The data mapping between these two spaces is given by,

$$ {}^{ENU}\underline{\mathbf{p}} = {}^{ENU}\mathbf{q}_{\mathcal{S}(t=0)} \star {}^{\mathcal{S}}\underline{\mathbf{p}}(t) \star {}^{ENU}\bar{\mathbf{q}}_{\mathcal{S}(t=0)} \tag{2.77} $$

recalling that ${}^{\mathcal{S}}\mathbf{p}(t)$ is a vehicle point fix determined by visual odometry, ${}^{ENU}\mathbf{p}$ denotes the corresponding coordinates in the ENU frame and ${}^{ENU}\mathbf{q}_{\mathcal{S}(t=0)}$ is the quaternion rotation representing the unknown attitude.

Since the GPS positioning points in the ENU frame must match the visual odometry ones in the frame $\mathcal{S}(t=0)$, the optimal rotation ${}^{ENU}\mathbf{q}_{\mathcal{S}(t=0)}$ can be found as a least squares solution. Based on the Singular Value Decomposition (SVD) of the correlation matrix of the centered point sets, this rotation is given by [AHB87]:

$$ {}^{ENU}\mathbf{q}_{\mathcal{S}(t=0)} = \mathbf{q}(\mathbf{V}\mathbf{U}^T) \tag{2.78} $$

with

$$ \mathbf{U}\mathbf{S}\mathbf{V}^T = [{}^{\mathcal{S}}\mathbf{p}(0,...,i,...,t) - {}^{\mathcal{S}}\bar{\mathbf{p}}]_{3\times t}[{}^{ENU}\mathbf{p}(0,...,i,...,t) - {}^{ENU}\bar{\mathbf{p}}]^T_{t\times 3} \tag{2.79} $$

where the correlation matrix of the centered point sets is a 3×3 matrix which is factored into left and right orthogonal matrices $\mathbf{U}$ and $\mathbf{V}^T$ and a diagonal matrix $\mathbf{S}$. ${}^{\mathcal{S}}\mathbf{p}(0,...,i,...,t)$ is the set of fix estimates provided by the visual odometry, ${}^{\mathcal{S}}\bar{\mathbf{p}}$ is the centroid of the point set ${}^{\mathcal{S}}\mathbf{p}(0,...,i,...,t)$, ${}^{ENU}\mathbf{p}(0,...,i,...,t)$ is the point set of GPS associated fixes and ${}^{ENU}\bar{\mathbf{p}}$ is its centroid.

This method is quite easy to implement and provides a reliable heading most of the time. It presents, however, some lacks when the initialization trajectory represents a straight line in space. In this case, all the degrees-of-freedom are not constrained, and hence, the estimated heading is a singular solution. Moreover, this method is not robust to discontinuous or inaccurate GPS initialization trajectory. These special cases can be coped with the use of a Geographical Information System (GIS) and a map matching routine.

Geodetic coordinates of the ego-vehicle

This routine is intended to perform an on-line conversion of the visual odometry estimates of the ego-vehicle into a GPS like positioning. It consists in computing, consec-

utively, the geodetic coordinates, ${}^{LLH}\mathbf{p}$, of the local vehicle position, ${}^{\mathcal{S}}\mathbf{p}(t)$, once the initialization step is finished. Thus,

$$
\begin{aligned}
{}^{ENU}\underline{\mathbf{p}} &= {}^{ENU}\mathbf{q}_{\mathcal{S}(t=0)} \star^{\mathcal{S}} \underline{\mathbf{p}}(t) \star^{ENU} \bar{\mathbf{q}}_{\mathcal{S}(t=0)} && (2.80)\\
{}^{ECEF}\underline{\mathbf{p}} &= {}^{ECEF}\mathbf{q}_{ENU} \star^{ENU} \underline{\mathbf{p}} \star^{ECEF} \bar{\mathbf{q}}_{ENU} + {}^{ECEF}\underline{\mathbf{t}}_{ENU} && (2.81)\\
{}^{LLH}\mathbf{p} &= ecef2llh({}^{ECEF}\mathbf{p}) && (2.82)
\end{aligned}
$$

with

$$
{}^{ENU}\mathbf{q}_{ECEF} = {}^{ECEF}\bar{\mathbf{q}}_{ENU} \tag{2.83}
$$

where $ecef2llh(\cdot)$ is an iterative function to convert Cartesian ECEF to geodetic WGS-84 coordinates.

2.8 Real-time 3D Visual Odometry Algorithm

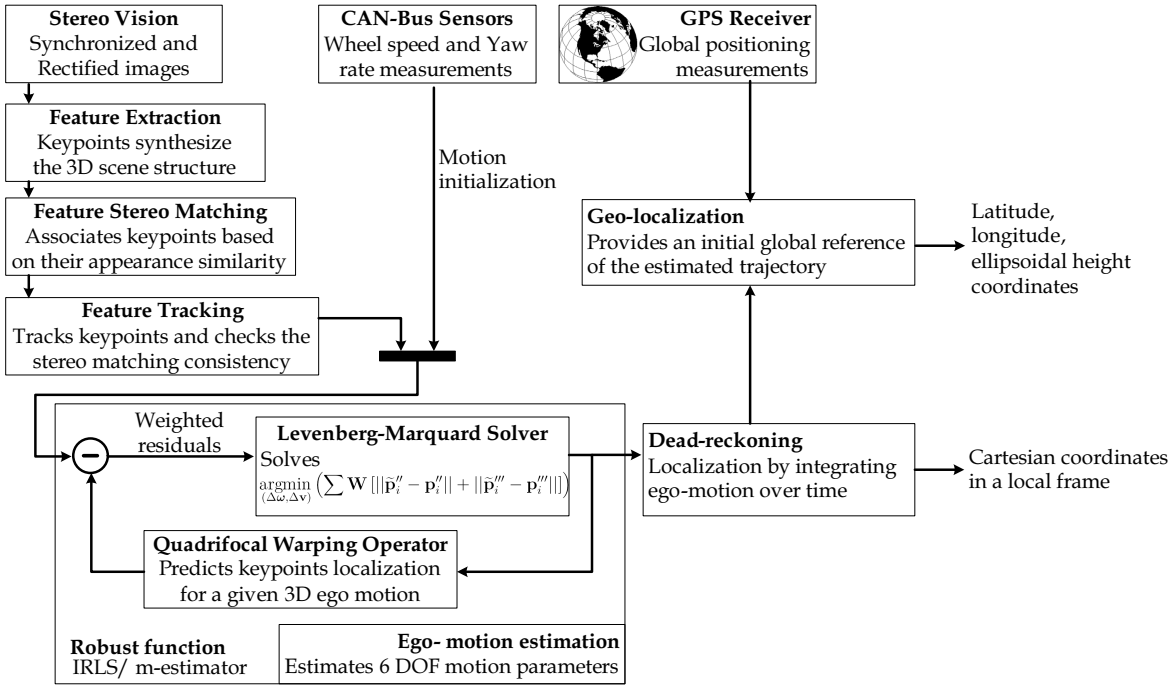


Figure 2.29: *Localization system overview*

In section 2.5, the sub-routines necessary to the vision-based odometry method have been detailed. An overview of the proposed strategy shown in Fig. 2.29 and the complete method is summarized in the algorithm 2.8.5. As it has been indicated, the proposed algorithm has been developed considering a calibrated stereo vision system which provides rectified images. The main objective is to estimate, from succeeding

stereo image pairs, the ego-motion parameters representing the 3D trajectory followed by the SVS.

Algorithm 2.8.5 Real-time 3D Visual Odometry Algorithm

Input: - Synchronized and rectified gray-scale stereo images (320×240 pixels resolution)
- Intrinsic and extrinsic parameters of the stereo vision system
- Keypoint detector, typically SURF features
- Template correlation size (usually defined for 9×9 and 11×11)
- Dynamic disparity interval
- Epipolar alignment error tolerance (typically ± 2 pixels)
- ZNCC minimum acceptable value (typically 0.7 to 0.8)
- Pyramid levels (typically 3 levels)
- Min. tracked keypoints for motion estimation, typically 30 feature points are required
- Prior motion estimate ${}^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega}_0,\ \Delta\mathbf{v}_0]_{\mathcal{S}(t)} \in \mathbb{SE}(3)$
- IRLS maximum iteration cycles (typical value 8)
- Levenberg-Marquard maximum iteration cycles (typical limit 60)

Optional inputs:
- Planar motion (typically provided by WSS-Gyro measurements)
- Global positioning measurements

Output: - 6 degrees-of-freedom (DOF) Ego-motion parameters ${}^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}]_{\mathcal{S}(t)}$
- Locally referenced positioning in Cartesian coordinates

Optional outputs:
- Geo-referenced positioning in WGS-84 coordinates (latitude, longitude, altitude)

1: **while** Data acquisition **do**
2: ▶ Acquire a new rectified gray-scale stereo image pair
3: **if** Initialization routine **then**
4: ▶ Keypoint extraction from reference views ($\mathbf{I}^*$ and $\mathbf{I}'^*$)
5: ▶ Load left and right reference key feature sets (p^* and p'^*)
6: ▶ Associate keypoint sets $\{\mathsf{p}^*, \mathsf{p}'^*\}_s$ using the stereo matching algorithm 2.4.3
7: **end if**
8: **if** Tracking routine **then**
9: ▶ Track the reference stereo features $\{\mathsf{p}^*, \mathsf{p}'^*\}_s$ to the current stereo view using the pyramidal Lucas-Kanade algorithm

Begin IRLS Optimization routine:

10: **if** Enough stereo features are tracked **then**
11: ▶ Initialize the LM algorithm with a prior estimate ${}^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega}_0,\ \Delta\mathbf{v}_0]_{\mathcal{S}(t)}$
12: ▶ Initialize weights (e.g. typically $\mathbf{W} = \mathbb{I}$)
13: **repeat**
14: ▶ Solve

$$\underset{(\Delta\boldsymbol{\omega},\Delta\mathbf{v})}{\operatorname{argmin}} \left(\sum \mathbf{W} \left[||\tilde{\mathbf{p}}_i'' - \mathbf{p}_i''|| + ||\tilde{\mathbf{p}}_i''' - \mathbf{p}_i'''|| \right] \right)$$

15: ▶ Extract optimization residuals and update weights
16: ▶ Evaluate changes of the optimized motion parameters
17: **until** Optimized motion parameters are not stable AND
Maximum iteration limit is not reached AND
Outlier breakdown point < 0.5
18: **else**
19: ▶ Re-initialize algorithm, go to line 3
20: **end if**
End IRLS Optimization routine

21: **end if**
22: Swap all $t-1$ and t variables
23: **end while**

2.9 Experiments

Several experiments have been carried out to validate the proposed 3D visual odometry method and to analyze and quantify the accuracy of the method with respect to other sensing techniques. Following the description presented in section 2.5, the 3D visual odometry method was implemented in C/C++ using Intel's OpenCV toolbox [Ope, BK08] and the Levenberg-Marquardt nonlinear least squares algorithm implementation, *levmar,* [Lou].

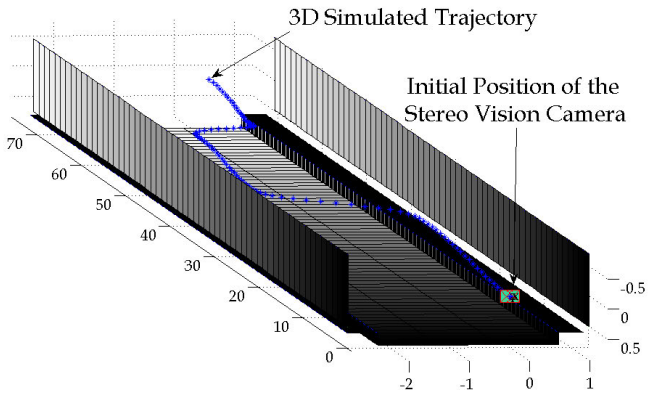


Figure 2.30: *SVS trajectory in a simulated scene*

In the sequel, experimental results of a first test under simulated conditions and two other tests using real data are presented.

2.9.1 Simulation

During the development of this work, it was not so easy to find an error function which lets us calculate a good motion estimation by ensuring an affordable time of convergence for real time execution. For this, we started by carrying out a test using data that simulates quasi-urban conditions, such as the acquisition frequency, the image resolution, the 3D inter-frame motion and the presence of static and dynamic objects.

The considerations which were taken into account in the simulation model correspond to a stereo pair image resolution of 320×240 pixels at an acquisition frequency of 15fps. In order to generate the synthetic stereo images (see Fig. 2.31, upper stereo image pair), the 3D camera motion was estimated as a function of the vehicle velocity curve and the camera acquisition frequency. The range of the applied 3D ego-motion varies between a minimum of 0.27 cm and a maximum of 0.92 cm. These conditions simulate respectively vehicle speeds between 15 Km/h and 50 Km/h along a 66 m trajectory (see Fig. 2.30).

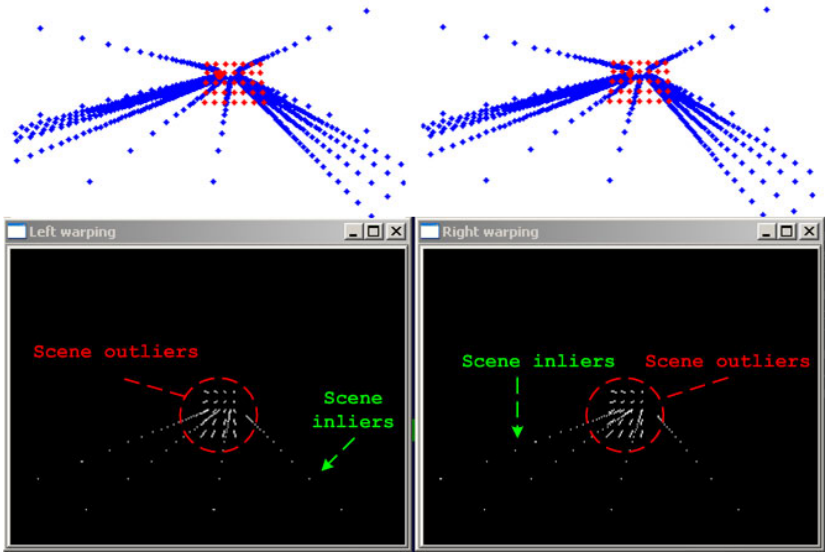


Figure 2.31: *Simulation test with outliers*

For the simulation test, a total of 100 visible simulated feature points were provided to the algorithm for each ego-motion estimation. A first trial have consisted in estimating the 3D trajectory followed by the SVS under ideal conditions. Thus, the features points provided to the algorithm corresponded to static points only, which were projected onto the image plane without noise and no stereo matching errors. In the upper section of Fig. 2.32 is illustrated the obtained trajectory results showing the quality of the optimized error function. The convergence of the LM algorithm was achieved even by initializing the ego-motion parameters to zero.

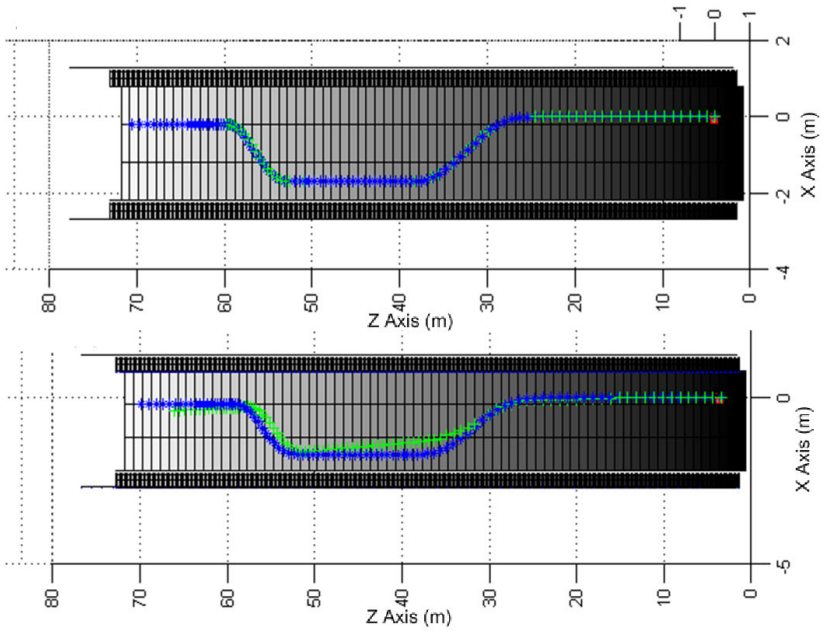


Figure 2.32: *Obtained trajectory in simulated conditions (Blue: Ground truth trajectory, Green: Estimated trajectory)*

A second trial was performed in order to validate the robustness of the algorithm by having 20% of the features coming from dynamic objects in the simulated environment, and adding some random stereo-feature mismatching and a one pixel σ noise. The obtained 3D trajectory is shown in the lower section of Fig. 2.32.

The results which are illustrated in the lower image pair of Fig. 2.31 show that the algorithm converges into a motion solution which minimizes the optical flow error generated by the static environment.

Fig. 2.33 shows the error time evolution computed along the obtained trajectory in the second trial. This curve lets us observe a drift, generated particularly in the trajectory sections where there is an important rotation motion (i.e. 90° turns). This is because important ego-motions need more iterations for convergence. Note that the number of iterations is constrained here because of real-time execution.

The error evolution curve illustrated in the upper plot of Fig. 2.33 shows the error percentage computed with respect to the traveled distance. The absolute positioning error shown in the lower plot of Fig. 2.33 is computed as the euclidean distance between the ground truth and the estimated fix. As expected, the absolute positioning error increases with distance as the estimation errors are continuously integrated. This constitutes the main drawback of *dead-reckoning* techniques. The simulation results are, however, meaningful for a first method validation.

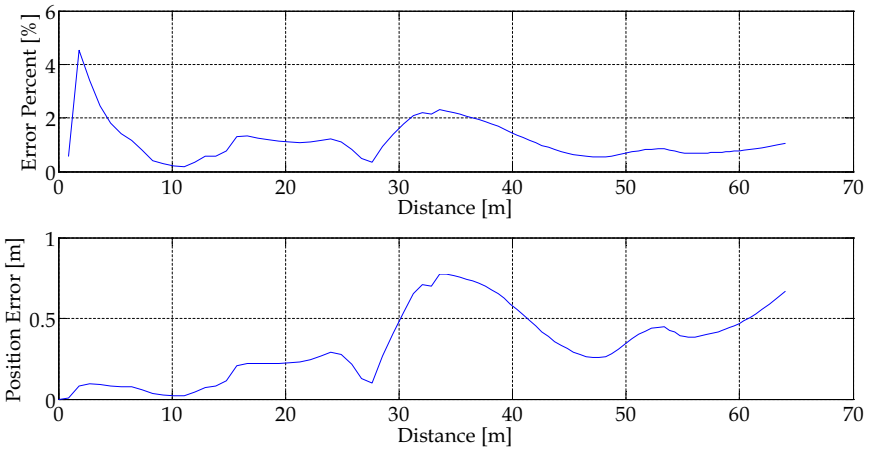


Figure 2.33: *Error evolution in simulated conditions*

2.9.2 Real data results

Experiments using real data were carried out thanks to the experimental vehicle Carmen of the Heudiasyc laboratory. In this section, two different trials using real data are reported. During both trials the vehicle was equipped with a 47cm-baseline Videre Stereo Vision System installed at the top. This system is composed of CMOS cameras with 4.5mm lenses that were set up to acquire gray-scale images with a resolution of 320×240 pixels.

The visual odometry algorithm was always set to extract up to 300 SURF keypoints and to track them over time. The reinitialization condition is triggered when there are no more than 30 features available for the ego-motion estimation. The IRLS optimization routine was limited to 8 iterations and the LM nonlinear minimization to 60 iterations for the real-time execution.

In order to compare the visual odometry results to other localization techniques, the data provided by a Septentrio PolaRx2c GPS receiver and the vehicle proprioceptive sensors (WSS-Gyroscope) were also accurately logged (i.e. time stamped data). Hereafter, the obtained results are presented and discussed.

Experimental trial in open loop trajectory

In the first trial, the vehicle was driven over a 790 m long trajectory in a quasi-urban environment characterized by moving objects (i.e. vehicles, cycles, pedestrians), some buildings and trees. The vehicle velocity was less than 60 Km/h. The video sequence was recorded at a frequency of 15 fps in sunny exposition conditions. During the experiment, GPS and proprioceptive data were also acquired simultaneously by the same multi-threaded application. The execution performance illustrated during this trial has been obtained on a Intel Core 2 CPU 2.1 GHz running under Windows XP

OS.

Figure 2.34: *Feature points during real-time odometry estimation,■: Outliers □ : Inliers*

In order to verify the results coherence regarding the stereo feature matching and the motion segmentation of the observed scene, the feature points were indexed and displayed in the stereo images. As shown in Fig. 2.34, the indexation reveals a good stereo matching performance, since almost all of them are correctly associated. The shown classification of the stereo features identifies the inliers, used during the ego-motion estimation, by empty squares. It is also worth to mention that almost all the stereo feature points lying in the moving vehicle are classified as outliers (i.e. filled squares) by the robust function.

As can be noticed in Figure 2.35, some objects like tree leaves are an important source of possible mismatching. This sort of objects increases the test scenes complexity. In spite of that, the rejection of moving objects has performed quite well.

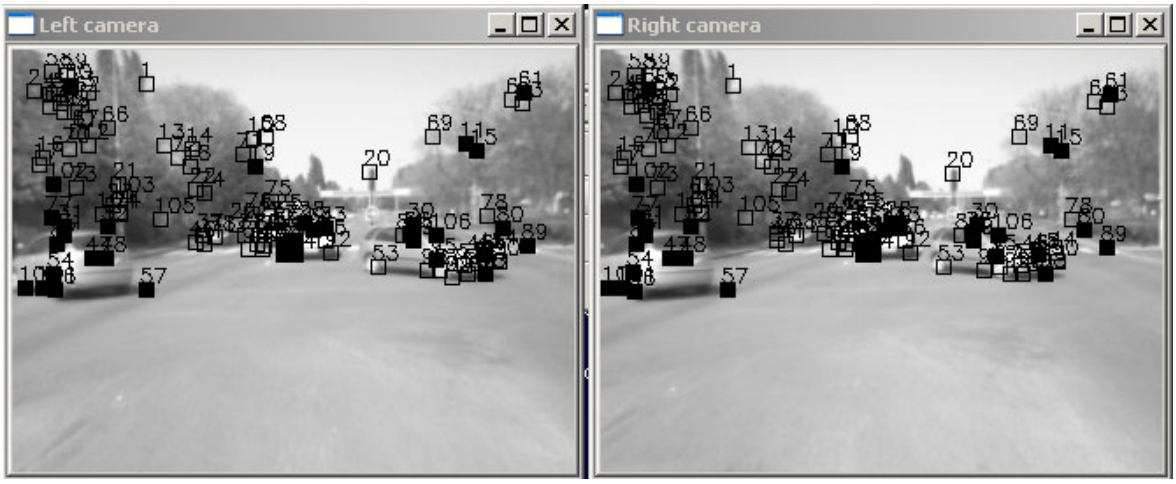


Figure 2.35: *Feature points lying in moving objects ■ : Outliers □ : Inliers*

No ground truth localization system was available during this experiment, particularly for the attitude estimates. Therefore, we only report here the comparison of the visual

odometry trajectory with the GPS receiver (Fig. 2.36 and Fig. 2.39) and the proprioceptive sensors (WSS-Gyro) 2D Odometry (Fig. 2.36). As the key characteristic of any dead-reckoning system is its drift with respect to the traveled distance, the low accuracy (absolute error) of the GPS used here is not an issue. Since the GPS was used in differential SBAS (Satellite Based Augmentation System) mode, the atmospheric effects that often bias the pseudo-ranges are subject to slow variation. So, the precision (relative error) is very good in open sky conditions (less than one meter typically). Therefore, using a GPS for comparison is meaningful, even if there are still multi-path effects and satellites outages that introduce jumps. It is important to keep in mind that this approach does not allow estimating the real errors, only differences are computed.

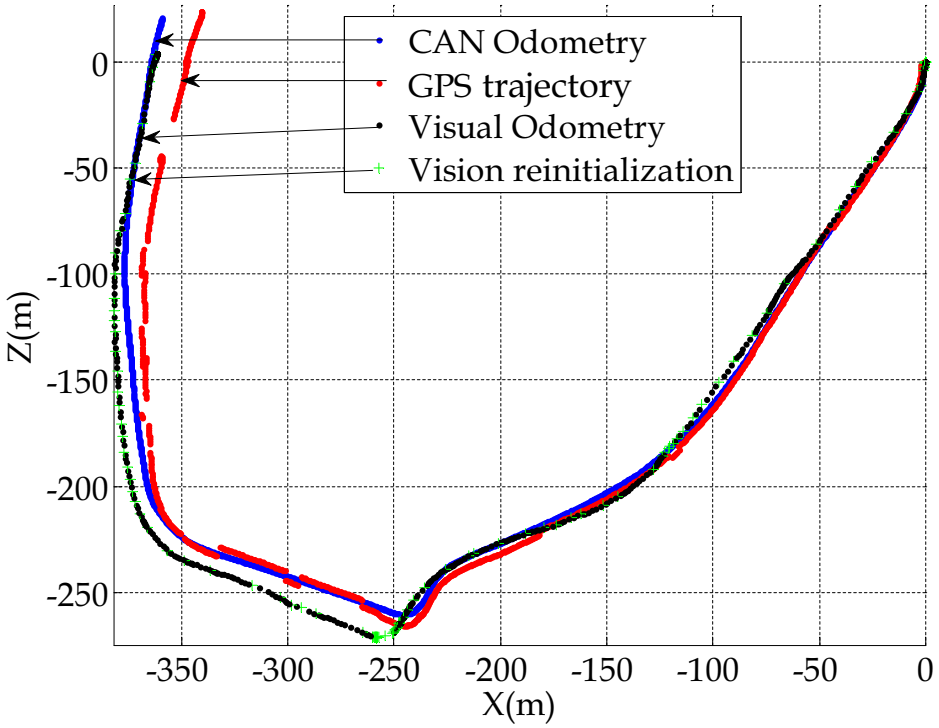


Figure 2.36: *2D estimated trajectory using GPS, proprioceptive sensors and visual odometry*

The visual odometry position was initialized using the first GPS point and the attitude using the median of the first 50 points headings as explained before. Taking into account the total trajectory distance of the trial, planar (x-z plane) and vertical (y-axis direction) drifts of 3.9% and 0.25% were obtained respectively, using the visual odometry. These differences correspond to the ratio of the Euclidean distance with respect to the GPS position over the total traveled distance. The 2D-odometry difference evolution is shown in the Fig. 2.37.

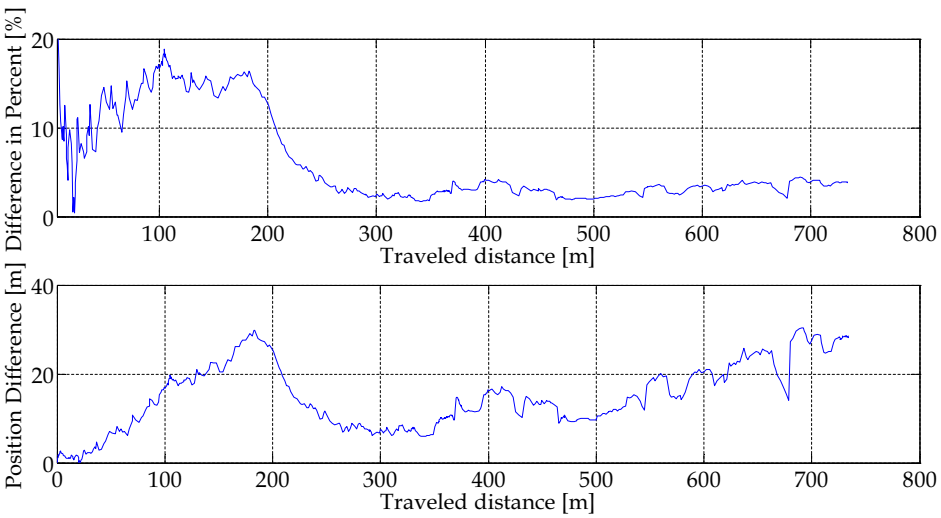


Figure 2.37: *Difference evolution between GPS and Visual Odometry*

These results highlight a quite acceptable visual odometry drift. Different causes can lead the visual odometry to drift. The principal cause occurs when, in the observed scene, the number of features lying in vehicles exceeds the feature points in the static scene. For instance, in case of a total or partial occlusion of the SVS field of view by a mobile object (i.e. more than 40% of the image), there might be more feature points on this object than in the static scene. Thus, the algorithm would estimate the vehicle motion with respect to the mobile object. Such situations break the principal assumption of the robust optimization process. This constitutes the main drawback of the sparse approach.

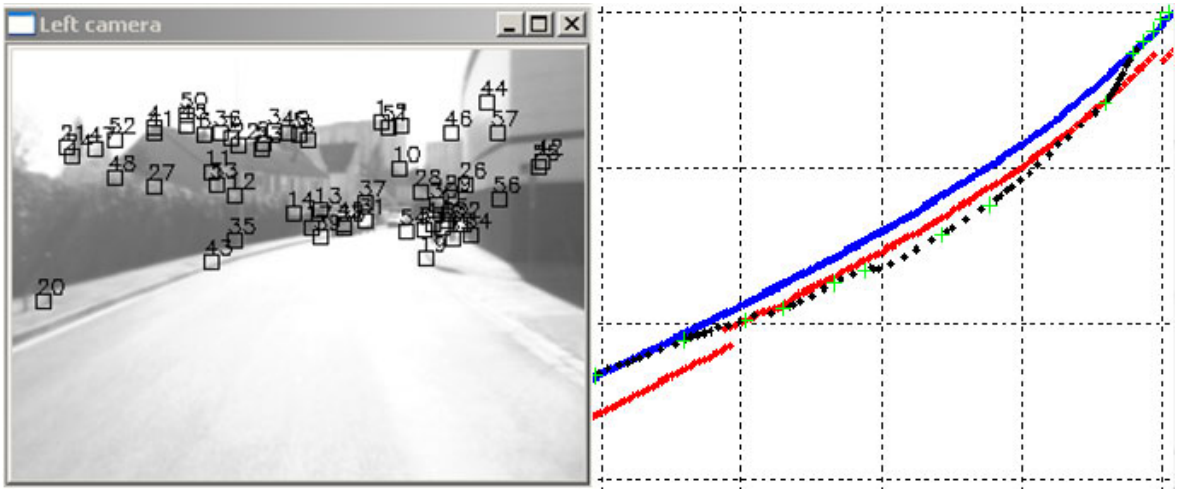


Figure 2.38: *Observed drift caused by critical conditions*

Another observed drift source is caused by the aperture problem in the tracking routine. Fig. 2.38 shows a zoomed part of the trajectory reconstructed in Fig. 2.36, highlighting a drift start. By observing the left stereo corresponding image, one can see that some feature points present a sliding motion (left sidewalk) and a few of them lie on poor

texture surfaces, such as the reflective pattern on the top-right building. These features induce a bias on the optical flow. In addition, the non-uniform feature dispersion on the image plane also decreases the precision of the motion estimation.

During the trial, each reinitialization of the algorithm was logged as well. These data allow us to detect situations where very few points were tracked in the process. Please note that the lack of tracked feature points represents a situation which did not appear in the simulated conditions. This phenomenon is a crucial issue in real conditions. Fig. 2.36 shows in green crosses when the visual odometry is reinitialized for re-extracting new feature points. As expected, the algorithm re-initializes frequently in turns because the observed feature points go easily out of view.

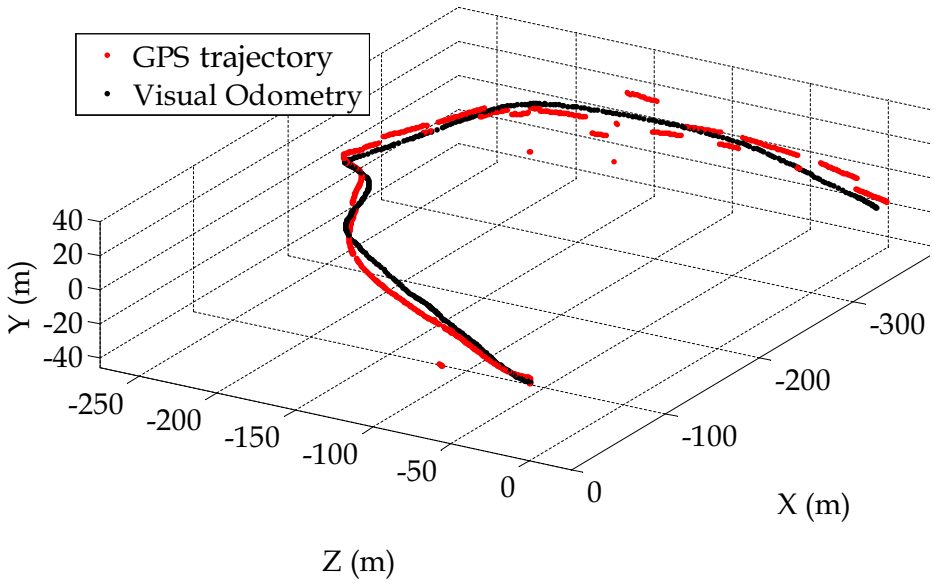


Figure 2.39: *3D estimated trajectory (local frame coordinates)*

Even if one can conclude that visual odometry does not improve the 2D accuracy compared to WSS odometry on good quality roads, it should be noticed that it provides the full 6 degrees of freedom which is not possible using wheel-based odometry.

Fig. 2.39 plots the 3D estimated trajectory obtained using visual odometry and GPS. The GPS jumps, particularly visible in this plot, are mainly due to the satellites changes. One can notice that the 3D trajectory, obtained using the visual odometry, is quite smooth. This is an interesting feature for integrity monitoring since a smoothed prediction of the pose is crucial to eliminate GPS outliers. Additionally, the altitude drift is very small (less than 0.3% of the traveled distance) which is a very good quality.

Finally, Fig. 2.40 presents a time execution histogram of the algorithm revealing that convergence time is not constant. This is caused by the non homogeneous vehicle motion and by the variability of scene complexity (i.e. outliers/inliers ratio). However, approximately half of the ego-motion estimations converged in less than 100 ms, this being a good real-time performance for a non optimized implementation.

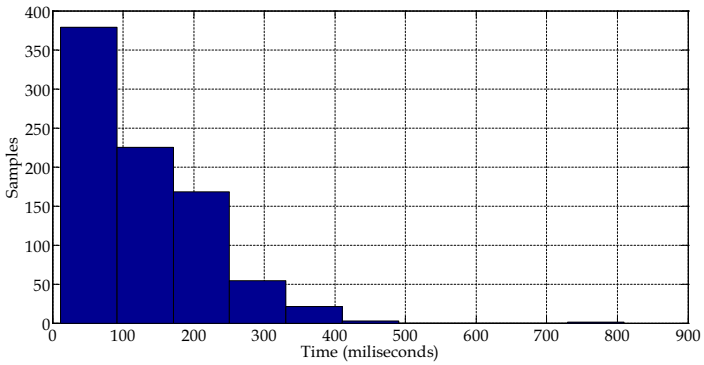


Figure 2.40: *Computation time histogram for the complete sequence*

Closed loop experiment

The goal of the second trial is the validation of the multi-modal odometry strategy (i.e. Visual Odometry + proprioceptive sensors), thus, the reported results were obtained using the technique described in section 2.6. The second trial conditions were different w.r.t. the first one, since the vehicle was driven in an urban environment in snowy conditions. The observed scenes were composed of low-rise buildings, trees and moving objects. The followed trajectory describes a 227m clockwise closed loop with pedestrians and vehicles. Low-textured scenes (e.g. rural environments and parking lots) were not included in this test. The vehicle speed was around 30 Km/h and the video sequence was recorded at 30 fps (i.e. maximum inter-frame distance of 27cm). As the image frequency acquisition has been increased, the data management required more computational resources. This is why an Intel Core Duo Quad CPU 3.1 GHz running under Windows XP OS was used to compute the reported results.

A first interesting result is illustrated in Fig. 2.41 where a set of feature points lying on a pedestrian are classified as outliers. This figure shows that even a small image motion induced by a pedestrian can be detected by the robust optimization scheme.

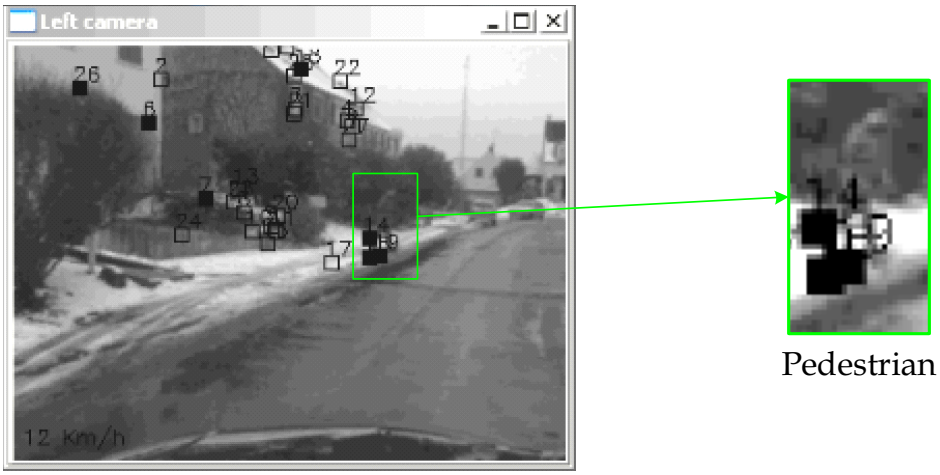


Figure 2.41: *Outlier rejection of keypoints lying on a walking pedestrian*

Two other examples of the rejection of the outlier feature points lying on moving vehicles are illustrated in Fig. 2.42. Since the outliers are principally located on moving objects, this information may be useful for a mobile object detector. However, such an application would require a dense feature tracking to ensure a continuous and reliable detection with a higher computational cost.

Figure 2.42: *Outlier rejection of feature points lying on moving vehicles*

Some complicated situations were observed during this test. Fig. 2.43 shows a 90° turn where just a few feature points are tracked. In addition, almost all of the tracked features are concentrated in the upper left image section providing a poor geometrical conditioning. The geometrical conditioning can not be improved, here, using a bucketing technique since half the image is poorly textured (i.e. road surface). This particular scene induced an important rotation error which was integrated all along the trajectory, thus increasing the final altitude drift significantly. This kind of situation can be avoided by a temporal filtering.

One can also highlight that even if the robust function can detect the stereo matching errors, it could also miss a few of them as shown in Fig. 2.43.

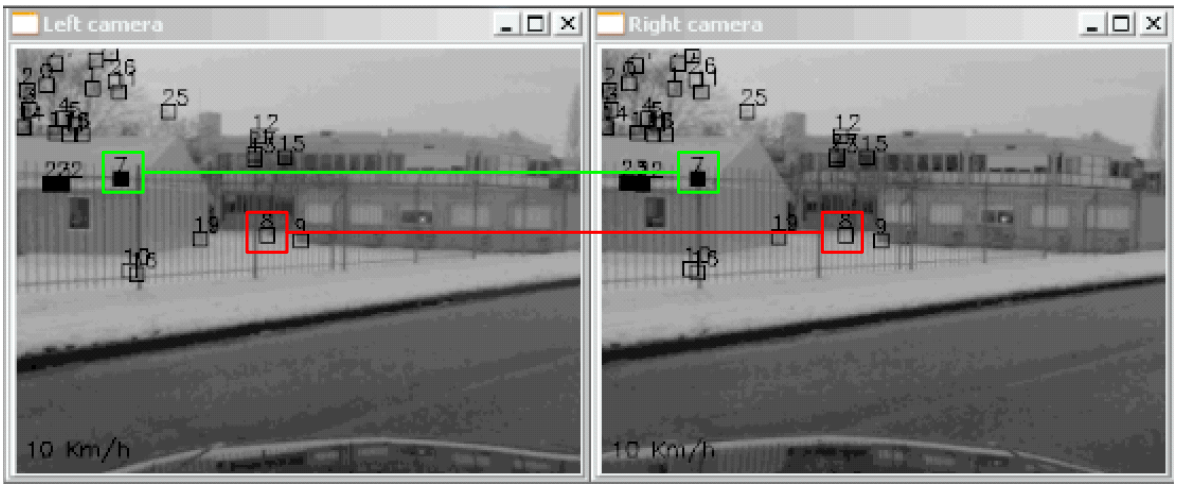


Figure 2.43: *Complex scene configured by a 90° turn*

The multi-modal 3D odometry trajectory, the WSS-Gyro based odometry and the GPS positioning were referenced in the same 3D space using the initialization method detailed in section 2.7. The 3D view of the complete closed loop trajectory is illustrated in Fig. 2.44.

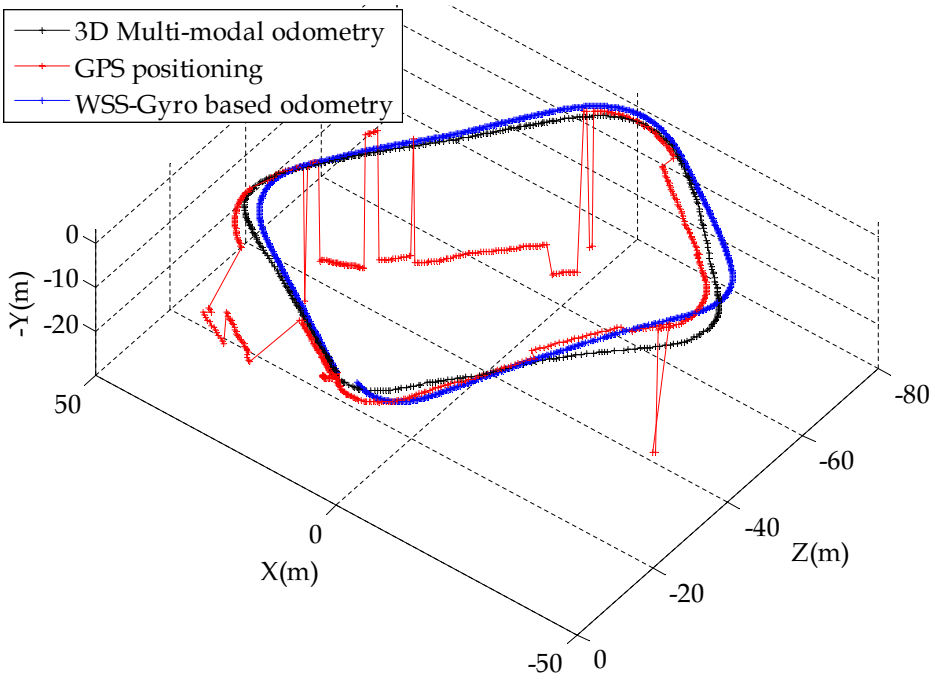


Figure 2.44: *Comparison of three different trajectories reconstructed from GPS, WSS-Gyro based odometry and multimodal 3D odometry. The 3D view of the reconstructed trajectories shows that multimodal 3D odometry is a complementary solution which can be used when GPS information is unreliable.*

The GPS performance during this trial was significantly decreased because of multi-paths and satellites outages. As shown in Fig. 2.44, the GPS positioning (i.e. red curve) was unreliable during half of the test sequence. A comparison between the GPS positioning and the multi-modal 3D odometry technique is provided through an aerial view of the estimated trajectories projections (see Fig. 2.45).

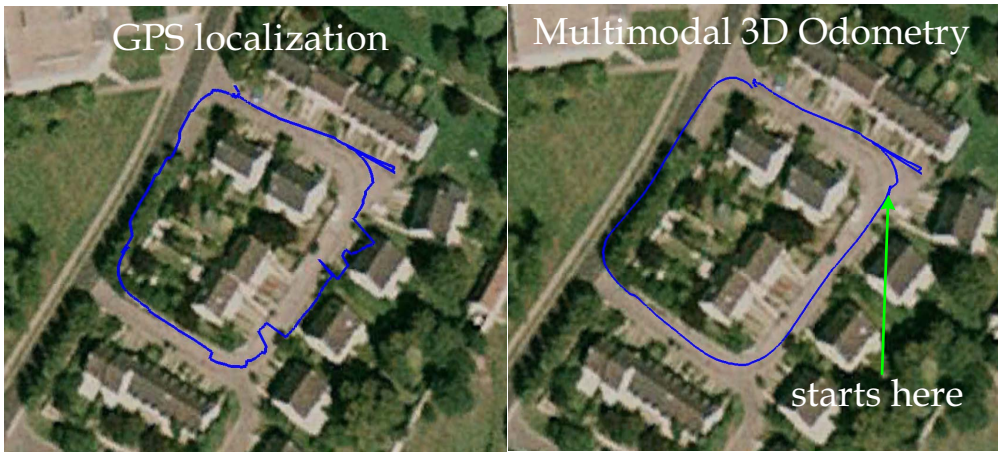


Figure 2.45: *Aerial view of the GPS and multi-modal 3D odometry trajectories projections*

From the figure, it can be verified that the precision of the GPS is reduced when the vehicle is close to the buildings. In the right side of the same figure, the multi-modal 3D odometry is Geo-localized. For this, the first segment of the GPS was used to initialize the algorithm. Once the system is initialized, the estimated vehicle fix is projected directly onto the map providing a smooth and reliable positioning.

The bird view of the estimated trajectories in Fig. 2.46, shows the gaps at the loop closing point. The 2D trajectory obtained using the WSS-Gyro based odometry (i.e. blue curve) achieves an acceptable drift representing 1.84% of the total traveled distance (i.e. 4.17 m for a closed loop). The total drift of the multi-modal strategy, computed as the Euclidean distance between the starting and the final trajectory position, was 1.9% of the total traveled distance. However, the drift of the planar trajectory projection of the multi-modal algorithm corresponds to 0.58% only, which represents an error 3 times lower than the one achieved by the WSS-Gyro based odometry.

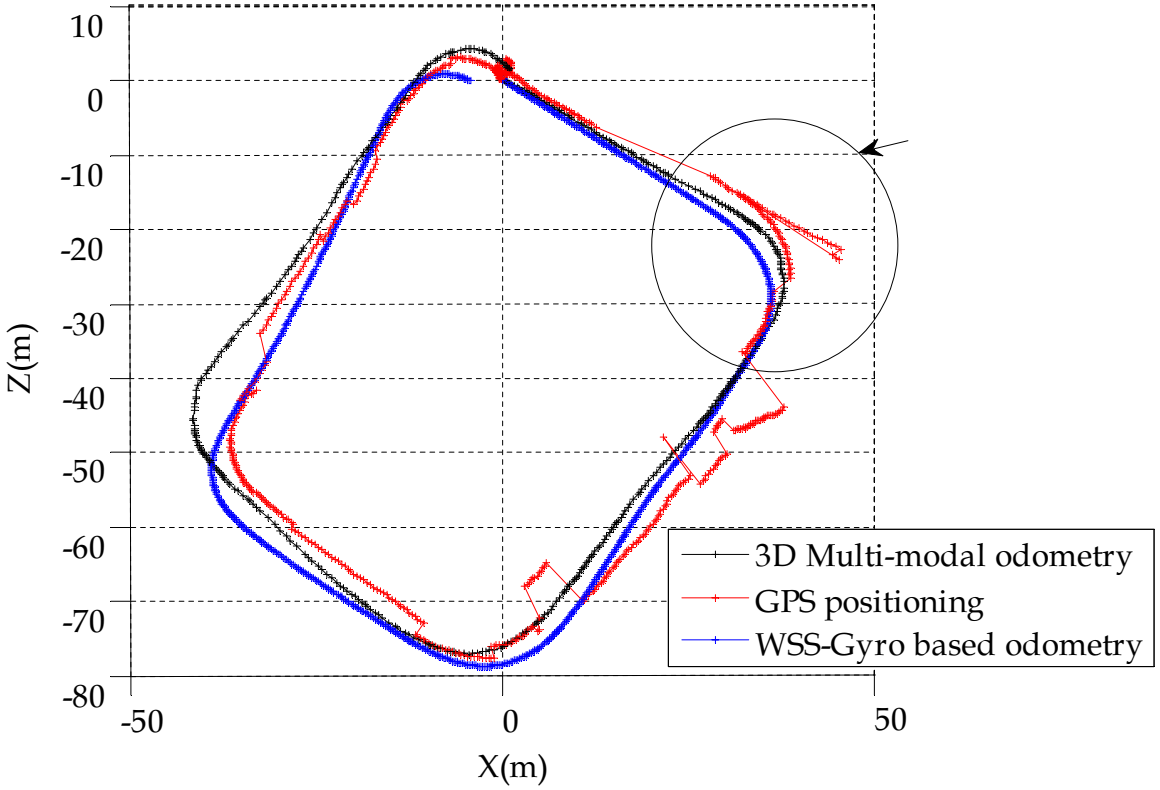


Figure 2.46: *The bird view of the 3D trajectory shows the improved results of the multimodal 3D odometry. The circled region evidences GPS jumps when visual odometry provides smooth estimates.*

The execution histogram which presents the convergence time of the multi-modal odometry algorithm is shown in Fig. 2.47. The computation time was reduced considerably, not only due to the more powerful computational resources but also thanks to the good initialization parameters provided by the planar odometry estimates. This initialization technique moderates as well the convergence variability observed during the first trial. The video sequence duration takes 90 seconds at 30fps providing 2700 stereo

pairs, where only about 2270 stereo pairs were processed. Some image pairs were skipped because the acquisition and the computation threads are independent. In this way, the implementation ensures that the algorithm receives the most recently acquired image pair.

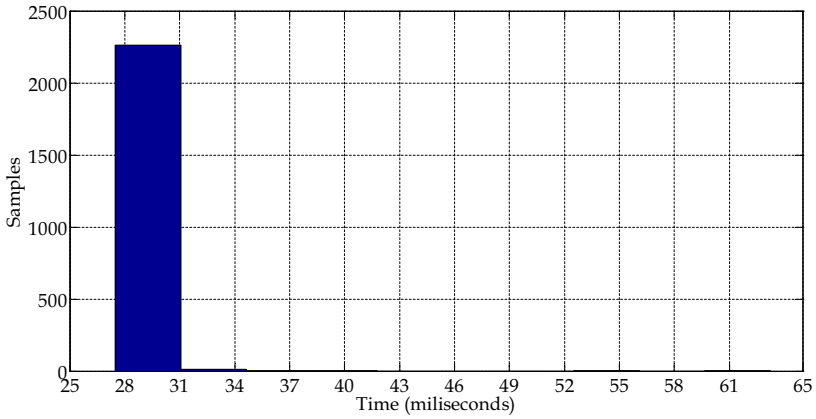


Figure 2.47: *Computation time histogram of the multi-modal odometry algorithm convergence along the complete video sequence*

In summary, these results show that the cooperative strategy helps coping with errors of the WSS-Gyro based odometry. Thanks to the planar motion initialization, which avoids any local minima solution, this technique improves the visual odometry performance in critical situations like high rotational speed in 90° turns and roundabouts. Additionally, the outlier rejection is enhanced and the minimization iteration cycles are reduced. The 3D ego-localization system performs quite well in situations when GPS can not provide a precise position.

2.10 Conclusion

A real-time 3D visual odometry approach has been presented and experimentally studied. The core of the method combines, in one non linear criterion, the ego-motion estimation based on sparse optical flow and the rigidity scene constraints provided by a quadrifocal tensor warping.

An experiment under quasi-urban conditions illustrates the good performance of the visual based only odometry with respect to GPS and proprioceptive ones. The obtained real-time results show a good trade-off between precision and execution time thanks to a sparse feature approach. This experiment revealed, however, that an important degradation source of the visual odometry performance is due to high rotational speed motions like 90° turns and roundabouts.

To mitigate the influence of these critical conditions and the reduction of the drift, a multi-modal 3D odometry strategy was proposed. For this, the visual based odometry

was tightly coupled with 2D odometry obtained through the use of proprioceptive sensors (WSS-Gyro). This multi-sensor technique was evaluated and the obtained results in a second experiment under urban conditions have confirmed the effectiveness of the approach. The proposed method has achieved, thanks to closed loop trajectory test, 3 times less drift compared to WSS-Gyro based odometry. This multi-modal approach constitutes a complementary solution when GPS information is unreliable. The management of altitude drifts and the estimation of the localization uncertainty constitutes the main prospects of this research.

Chapter 3

Multi-Modal Object Localization and Tracking

Contents

3.1 Introduction

Advanced Driver Assistance Systems (ADAS) can improve road safety thanks to obstacle detection, warning and avoidance functionalities. In this context, the precise

knowledge of the existence and the location and speed of the surrounding mobile objects constitutes a key information. Since the whole scene perception is provided from a moving vehicle, this problem is not trivial and entails the necessity of a precise ego-motion estimation to achieve good performance.

In the literature, different approaches address the object localization and tracking problem. Robotics approaches can be used to model the static part of the environment [DWB06] and to simultaneously detect moving objects [WTH+07]. These approaches are principally intended to retrieve a detailed environment representation or an accurate location of the mobile perception platform at any time.

Another approach, more suitable for urban navigation, was presented by Leibe et al. in [LCCG07]. This method makes use of a stereo vision based strategy to obtain a 3D dynamic map using a Structure-from-Motion technique and image object detectors. Alternatively, a Multi-layer lidar (for short ML lidar) alone can be used to estimate the ego-motion and to detect mobile objects thanks to a dense 3D grid-map approach [MON09]. In [SHRE00],[BCG+09] and [LRGA05] real time sensor-referenced approaches (i.e. ego-localization was not considered) were examined and implemented using multi-sensor systems showing the complementarity of lidar and vision systems. Moreover, the high level reliability achieved by this multi-modal perception concept makes it suitable for automotive applications.

In the aim of tracking objects as they move in space, a world-centric approach presents interesting properties once the ego-localization is estimated accurately (typically up to 1 cm per speed unit in Km/h). The tracking performance can considerably be increased, since the dynamics of the mobile objects are better modeled. This approach also simplifies the tasks involving scene understanding and ADAS implementation.

Ego localization can be achieved using proprioceptive exteroceptive sensors [CNPC08]. Coupling stereo vision and inertial measurements, a precise vehicle pose estimation can be achieved in real time as already presented in section 2.5.

Object tracking for ADAS is still an active research domain. Indeed, urban environments are characterized by complex conditions: moving and static objects, mobile perception, varied infrastructures. Object representation [PT09, NKL08], association methods [SB00], motion model and tracking strategies [LHL08] are key points which have to be considered with a particular attention.

Consequently, this chapter presents the formalization, the examination and an experimental validation of a multi-modal object detection, localization and tracking system. To this end, the system makes use of vision, lidar and inertial sensors. The outline of this approach is as follows: Firstly, objects are detected using a multi-layer lidar that is simultaneously used to identify a zone of interest (called maneuvering window) which reduces the complexity of the object-track assignment process. Then, only the objects lying within the maneuvering window are considered and localized w.r.t. a fixed refer-

ence (i.e. world frame) by compensating the motion of the ego-vehicle. Subsequently, objects are tracked in this world frame. The vehicle pose is obtained through the proposed localization technique presented in section 2.8 which provides reliable and good enough estimates.

The outcome of the proposed system consists in a sparse environment representation composed of kinematic states of the surrounding objects and the ego-vehicle, arranged in a *dynamic* map. The integrity of this map is then increased by the means of stereo vision that confirms or not the existence of the tracked objects. This self-assessment functionality has to manage the asynchronicity of the different sensing modalities.

The chapter is organized in four major parts: Object tracking, visual confirmation, multi-rate information management and experimental results. Section 3.2, details object detection, modeling and tracking issues. The visual track confirmation technique is then presented in section 3.3. The methodology employed to deal with the non-synchronicity of the sensing functions is presented in section 3.4. Finally, section 3.5 provides experimental results of the proposed concept in real conditions.

3.2 Object Tracking

This section describes a multi-modal perception strategy, able to estimate the planar trajectory of the surrounding objects as they move in the scene. This is made possible by integrating the spatial information (i.e. range measurements) provided by the lidar, and refining, over time, the knowledge of the detected objects dynamics. This function is formalized as a Multiple Target Tracking (MTT) problem which is addressed through a state-space filtering of the kinematic object states.

Multiple-Target Tracking (namely object tracking in the following), is a sequential process where a set of objects of interest are "followed", in a specific region of the field of view (maneuvering window). For this, input data coming from a single or multiple sensing sources, are necessary. Before going further, the concepts of object and track have to be defined. An object is a subjective representation located in space from a single measurement. For the particular case of a lidar, the kinematic properties of an object are unknown. In contrast, a track defines the kinematic state (i.e. location and speed) of an object which has been observed multiple times.

The sensing information consists in a set of incoming objects, which parameters are measured and used for the update of existing tracks or for the creation of new ones. The state of a track can only be updated if it is associated with (or assigned to) a newly observed object. The track process cycle is ended by a prediction stage of the tracks states, to the arrival time of the next set of incoming objects.

The tracked objects states, provided by this multi-modal perception function, are incorporated in a fixed-referenced model of the surrounding environment. The proposed

perception process performs in a fixed reference frame, since the precise mobile platform pose is estimated by visual odometry as detailed in section 2.5 of chapter 2.

The tracking strategy is outlined in five major parts as illustrated in Fig. 3.1: Object detection, maneuvering window identification, track prediction, object-track association and updating.

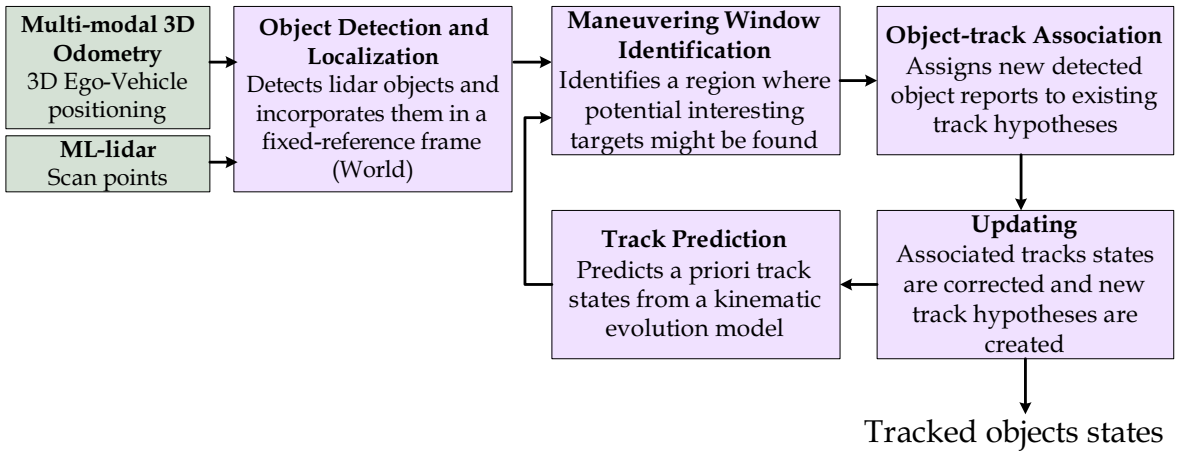


Figure 3.1: *The Overall Object Localization and Tracking Strategy consists in an iterative filtering scheme structured in five stages: object detection, maneuvering window identification, track prediction, object-track association and updating. This strategy makes use of lidar scan points and visual odometry and provides in output a set of tracked objects referenced to the world frame.*

3.2.1 Object Detection and Localization

Object detection is the first step that grants access to an automatic scene analysis. This process consists basically in a segmentation of the range measurements provided by the ML-lidar. Several works have addressed object detection using laser range data, coupled with classification algorithms, as part of the association process. In [NKL08], geometric predefined features combined with a Douglas–Peucker algorithm [NMTS07] (split-n-merge) have been used to detect and classify objects. This technique, however, requires prior object knowledge. Multi-modal techniques, performing a spatio-temporal segmentation of the sensing data, can also be used to provide an enhanced scene analysis [KZD04].

In this study, object detection does not make use of any classification step. Objects are dealt with in a generic context. The multi-layer scanner measurements are filtered, by discarding impacts which have been detected as lying on the road plane surface (filtering details are provided in section 3.2.2). The filtered ranging measurements of each ML-lidar scan are then segmented (clustered), using a technique similar to the one proposed by Dietmayer et al. [DSS01]. In this way, the object detection function delivers a set of surrounding objects, based on a 3D Euclidean inter-distance clustering. The output of this perception function is illustrated through a bird-eye view of the

segmented scan in the right-hand side of Fig. 3.2 with its corresponding scene image in the left-hand side.

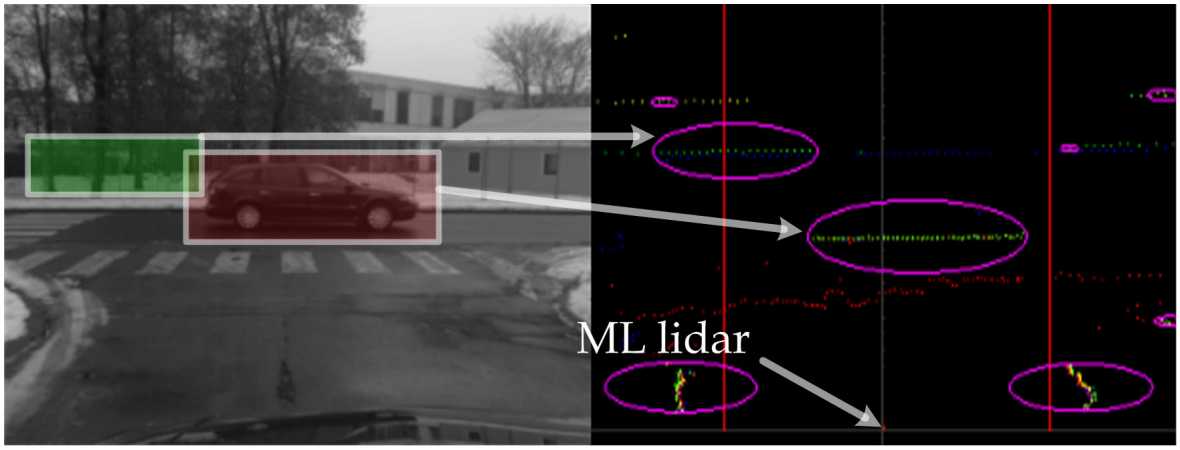


Figure 3.2: *The ML lidar-based object detection delivers a set of surrounding objects, based on an efficient 3D Euclidean inter-distance clustering. A bird-eye view of the segmented scan is illustrated in the right-hand side and the corresponding detected objects are referenced in the left-hand side image.*

The detected objects are characterized by their planar location (i.e. ${}^{\mathcal{L}}Z = 0$) in the lidar frame $\mathcal{L}$, and their dimension (i.e. bounding circle) as depicted in Fig. 3.3. Even if the multi-layer technology provides a vertical FOV (i.e. 3.2° coverage) of the scene, its 4-layer resolution is insufficient to observe objects heights. This is why the object size is computed from an orthogonal projection of the 4-layer data onto the plane ${}^{\mathcal{L}}Z = 0$. The circle bounding the object can be extended to a cylinder using a given object height.

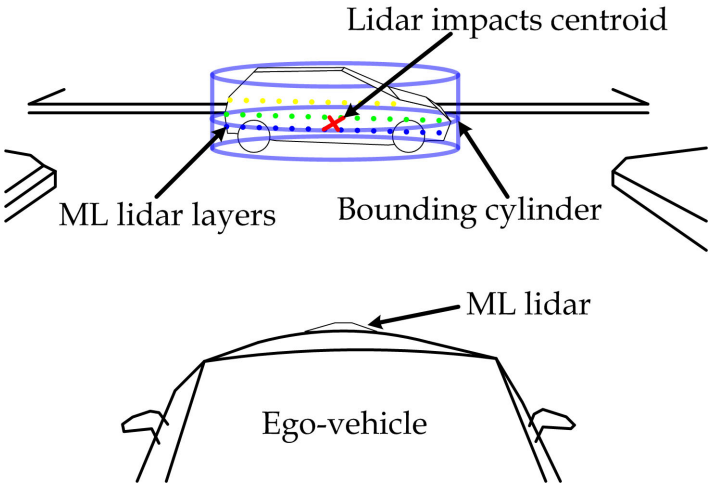


Figure 3.3: *The object model representation (in space) of the ML lidar objects is defined by a bounding circle of the segmented scan points and its planar location (i.e. ${}^{\mathcal{L}}Z = 0$) in the lidar frame $\mathcal{L}$. The bounding circle object representation can, be extended to a cylinder using a given object height.*

The imprecision and the uncertainty of each detected object are modeled through a

global detection confidence indicator, which relies on the following criteria [Fay09]:

- The ability of the ML lidar sensor to detect vertical objects with respect to its range position.
- The beam divergence, which worsens the measurement precision, particularly in situations of a non perpendicular incidence angle.
- The theoretical maximum number of laser impacts (per layer) lying on a detected object. This factor can be computed as a function of the object dimension, the detection range and the laser scanner resolution.

For the implementation details of these criteria please refer to [Fay09, p. 124-127].

As the objects are detected by the ML-lidar, their reported positions are locally referenced with respect to the lidar frame $\mathcal{L}$, which is rigidly linked to the mobile platform. The object motion in space, observed from a moving platform, follows, most of the time, a non linear behavior. This phenomenon is particularly noticeable in turns as depicted by Fig. 3.4 the observed trajectory of a pedestrian in a crosswalk.

Often, this non linear motion is frequently linearized, inducing important errors in the tracking process. Such errors can be reduced if the object motion is isolated from the moving platform motion: The tracking performance is increased by modeling the dynamics of the mobile objects with respect to a fixed-reference frame. This relies on a precise object localization of the ego vehicle.

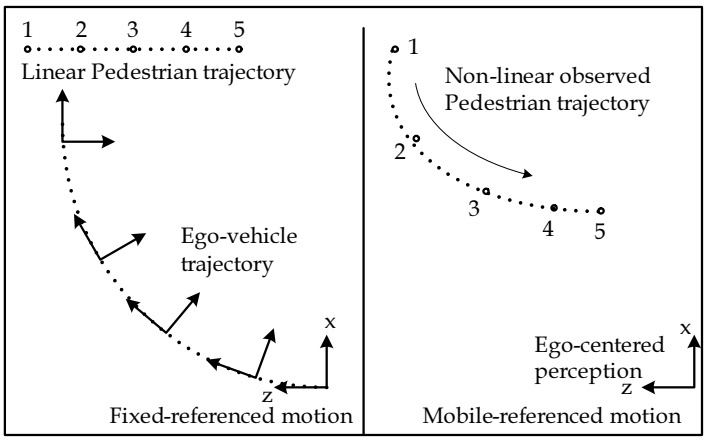


Figure 3.4: *Example of the observed motion of a pedestrian in a crosswalk while the ego-vehicle turns, using a fixed-reference (left) and a mobile referenced approach (right).*

Given the above, the proposed strategy consists in performing the tracking process in a fixed frame. Reported object fixes are transferred into a world frame $\mathcal{W}$ for further processing, by compensating the induced motion (3D ego-motion) of the mobile platform.

For instance, let ${}^{\mathcal{L}}\widetilde{\mathbf{y}}(t) = [x \;\; y \;\; 0]^T$ be the coordinates of a detected object at time t in the lidar frame. Its corresponding localization in $\mathcal{W}$ can be computed as follows, by composing two rigid transformations:

$$ {}^{\mathcal{S}(t)}\underline{\widetilde{\mathbf{y}}}(t) = \left({}^{\mathcal{S}}\mathbf{q}_{\mathcal{L}} \star^{\mathcal{L}} \underline{\widetilde{\mathbf{y}}}(t) \star^{\mathcal{S}} \bar{\mathbf{q}}_{\mathcal{L}}\right) +^{\mathcal{S}} \underline{\mathbf{t}}_{\mathcal{L}} \tag{3.1} $$

$$ {}^{\mathcal{W}}\underline{\widetilde{\mathbf{y}}}(t) = \left({}^{\mathcal{W}}\mathbf{q}_{\mathcal{S}(t)} \star^{\mathcal{S}(t)} \underline{\widetilde{\mathbf{y}}}(t) \star^{\mathcal{W}} \bar{\mathbf{q}}_{\mathcal{S}(t)}\right) +^{\mathcal{W}} \underline{\mathbf{t}}_{\mathcal{S}(t)} \tag{3.2} $$

where ${}^{\mathcal{S}(t)}\widetilde{\mathbf{y}}(t)$ and ${}^{\mathcal{W}}\widetilde{\mathbf{y}}(t)$ are respectively the coordinates of the detected object in the SVS and in the world frame at time t. The transformation denoted as ${}^{\mathcal{S}}[\mathbf{q},\ \mathbf{t}]_{\mathcal{L}}$ refers to the multi-sensor calibration between the SVS and the ML lidar frames which has been determined through the extrinsic calibration process. Additionally, ${}^{\mathcal{W}}[\mathbf{q},\ \mathbf{t}]_{\mathcal{S}(t)}$ represents the vehicle pose to be compensated in order to localize the objects in the world frame. This transformation[1] denotes implicitly the vehicle positioning at t in $\mathcal{W}$. The object localization process is summarized in Fig. 3.5.

Only the detected objects which are included in a zone of interest are localized w.r.t. the world frame using Eq. 3.1 and 3.2. This zone is dynamically detected and used as a global tracking gate as detailed in the following.

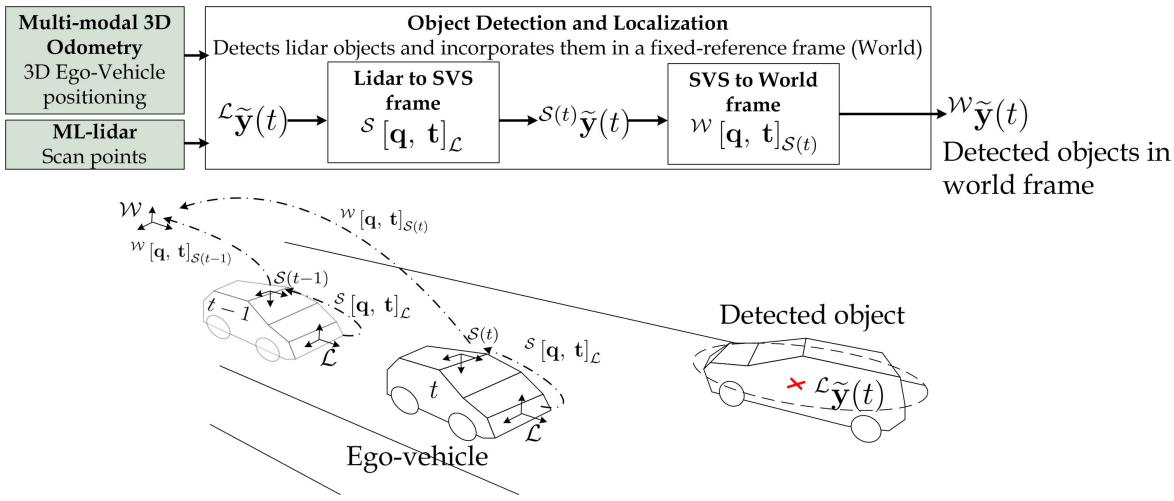


Figure 3.5: *The localization of the detected objects in the world frame is done by compensating the vehicle motion. For this, the coordinates of the detected objects are firstly transferred to the SVS frame making use of the extrinsic calibration parameters. Then, the object is localized in the world frame compensating the estimated vehicle motion.*

3.2.2 Maneuvering Window

Urban environments are complex, dynamic and completely variable. As such environments can be characterized by the presence of a large amount of static and mobile objects, the computational efficiency of an object tracker could be considerably decreased.

[1] In a further section, the detailed computation of this rigid transformation will be provided.

A key issue here consists in determining objects which constitute potential maneuvering targets and so computational resources can be dedicated mainly to tracks requiring a particular attention. Moreover, many incorrect observation-to-track pairings can be avoided. In the literature, the concept of maneuvering window has been widely studied in aeronautical MTT[2] applications [BP99]. This concept has been recently proposed for pedestrian detection applications in the LOVe project (French acronym for *Logiciels d' Observation de VulnérablEs* 2006-2009) [LOV], where this zone is estimated based on the ego-vehicle state (typically speed and steering angle).

For this study, the 3D observation space is restricted to a Zone Of Interest (ZOI) which is identified based on the prior knowledge of the scene only. This function provides the tracking algorithm with the ability to efficiently focus the computational resources on a maneuvering zone, where collisions might be predicted at appropriate reaction delays. Fig. 3.6 provides an example of the maneuvering window concept in an urban environment.

This function was proposed and implemented in real time by Fayad et al. in [FC07]. The method relies on maxima detection of lidar scan histograms.

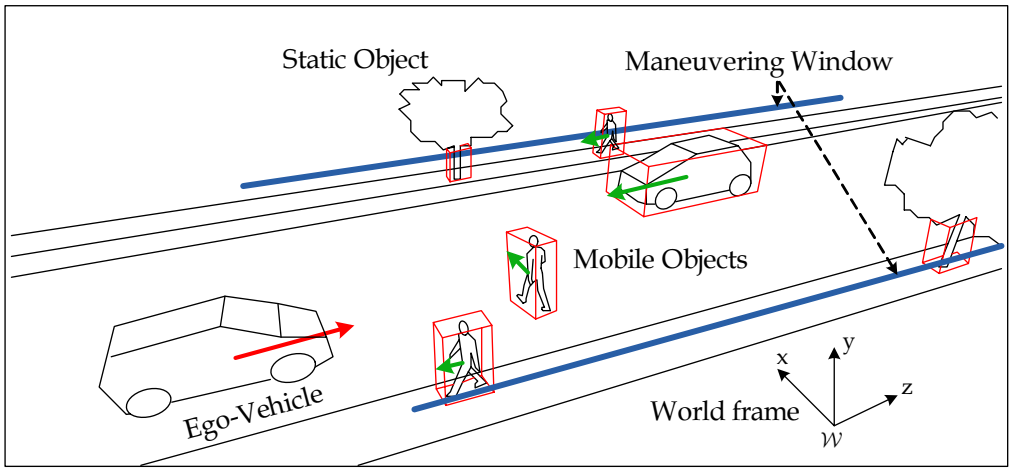


Figure 3.6: *Maneuvering window: This zone of interest aims at determining which objects constitute potential maneuvering targets. In this way, the computational resources can be appropriately dedicated to tracks requiring a particular attention and some first unlikely observation-to-track pairings can be avoided.*

A first lidar data filtering is performed not only to improve the ZOI detection but also to significantly decrease the scene clustering issues. This filtering consists in detecting the 3D lidar data corresponding to the road surface. This detection is done using the characteristic pattern observed when the two lower layers intersect the road plane at different angles as illustrated in Fig. 3.7. The second lidar layer can be predicted based on the first lidar layer and the geometrical constraints. The measurements belonging to the road plane are detected and excluded from further processes, when the Euclidean

[2] *Multiple Target Tracking*

error between the predicted and the measured layer is lower than a predefined threshold. The pitch angle of the ML-lidar noted ρ, is estimated by a temporal filtering and is updated using the detected road impacts. The parameters h (ML-lidar height) and γ (inter-layer angle) are considered known.

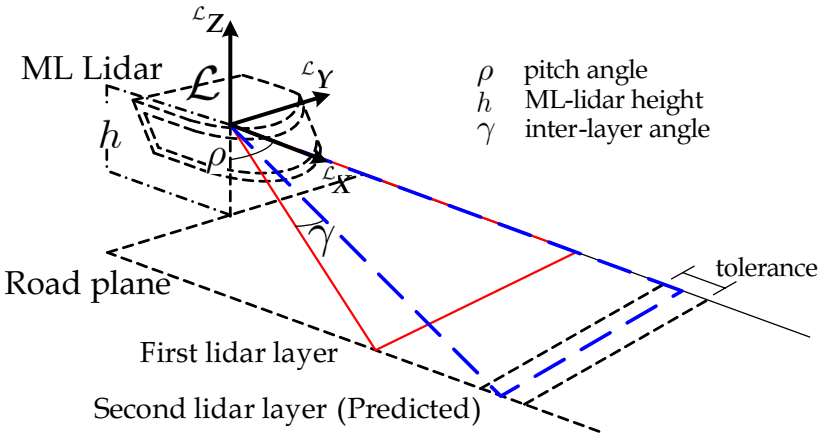


Figure 3.7: *Characteristic pattern observed when two layers of the ML lidar intersect the road plane. This geometrical constraint is used to identify which scan points lie in the road plane and to exclude them from further processes.*

The maneuvering window is characterized by two local limits in the x-axis direction of the lidar frame. As illustrated in Fig. 3.8, a 4-layer data scan is projected onto the ${}^{\mathcal{L}}XY$ plane (see the upper subplot) and provides an easy- to-exploit histogram ${}^{\mathcal{L}}Y$ axis (see the lower subplot). Objects like security barriers, walls and parked vehicles efficiently reduce the maneuvering window. The detected limits are finally filtered using a fixed-gain Luenberger observer in order to reduce the oscillations produced by important pitch changes situations [FC07].

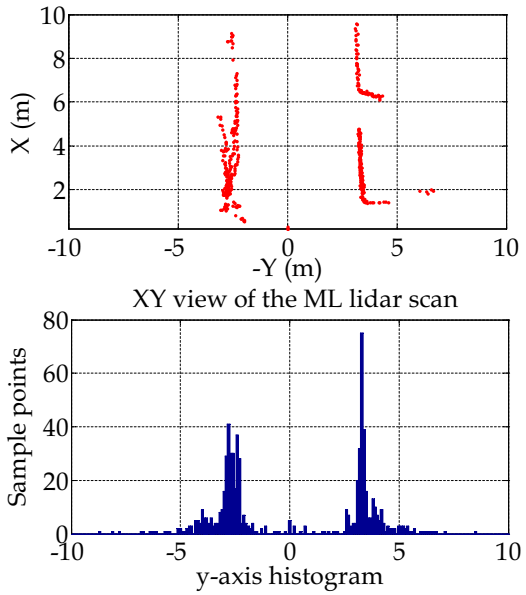


Figure 3.8: *Maneuvering window identification using y-axis histogram of the ML lidar scan points*

In turns and roundabouts scenarios, histogram peaks may fade out as illustrated in Fig. 3.9. In this case, the predicted localization of the maneuvering window limits would not be associated with new observations. Thus, the updating stage of the fixed-gain filter is not carried out, keeping in consequence the last maneuvering window estimation.

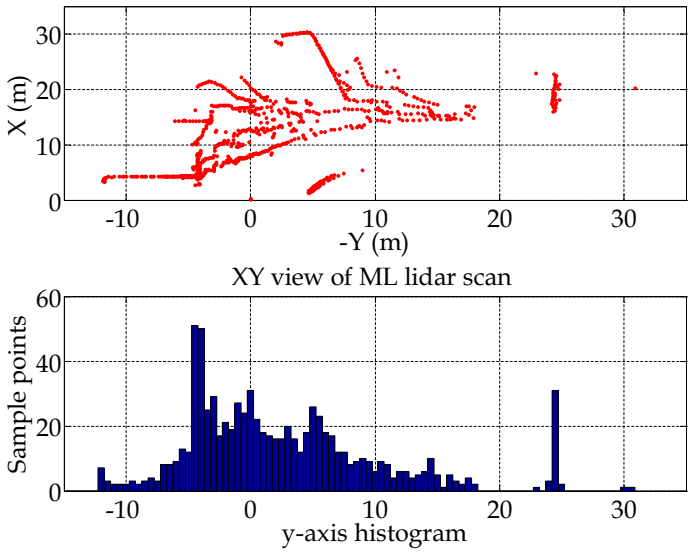


Figure 3.9: *ML lidar y-axis histogram peaks in turns may fade out. To deal with this, the detected limits are filtered using a fixed-gain Luenberger observer.*

3.2.3 Kinematic Track State and Evolution Model

Once the detected objects which are included in the maneuvering window are localized in the world frame, the tracking system can initialize the kinematic space state representations of these objects. Such a representation is known as "object state" and consists in a set of attributes arranged in a vector. The state is associated with its imprecision represented by a covariance matrix. As the same object is observed several times, the information, provided by multiple measurements over time, is integrated and fused in a "track state" with its own uncertainty allowance. These representations are detailed in the following.

Object and Track State Representations

The choice of the object and the track state representations mainly depends on the sensing characteristics and the tracking strategy. This choice has to be done relying on the parameters that have to be observed. This last statement implicitly assumes the state observability (observability issues are beyond the scope of this study). Hereinafter, two possible object and track state representations are considered.

A first possible tracking strategy consists in a tightly coupled scheme, where the track and the vehicle dynamics are stacked in the state vector. This approach is only well

suited for synchronous systems, since the object and the vehicle dynamics are constrained to be sampled (or observed) at the same time in order to be fused. It leads, however, to replicating the vehicle dynamics measurements for each track state, which is not computationally efficient at all.

A second tracking strategy consists in a loosely coupled fusion scheme, where the tracks and the vehicle dynamics are managed independently. This scheme is adopted in this study, since it allows the management of the sensing asynchronicity.

Consequently, the object state, for the loosely coupled MTT scheme, is defined as the localization of the detected object. Assuming that the objects included in the maneuvering window locally follow a planar motion, the object state can be efficiently reduced to the ${}^{\mathcal{W}}XZ$ components of its Cartesian coordinates in the world frame. With abuse of notation the object state is represented by ${}^{\mathcal{W}}\tilde{\mathbf{y}}(t)_{(x,z)}$. The uncertainty of the lidar object localization is modeled through a zero-mean, white, Gaussian measurement noise with an assumingly known covariance $\mathbf{N}$.

The track state is denoted by ${}^{\mathcal{W}}\mathbf{y}(t)$ and is composed of the ${}^{\mathcal{W}}XZ$ plane coordinates, $(x(t),\ z(t))$ in meters, and the planar velocity $(v_x(t),\ v_z(t))$ in $m \cdot s^{-1}$ stacked in a 4D vector as follows:

$$ {}^{\mathcal{W}}\mathbf{y}(t) = \begin{bmatrix} x(t) & z(t) & v_x(t) & v_z(t) \end{bmatrix}^T \tag{3.3} $$

This discrete notation is intended to represent the knowledge of ${}^{\mathcal{W}}\mathbf{y}(t)$ obtained by fusing all past reports up to time t. Additionally to the state parameters, other attributes are handled in parallel, like the object size in meters (i.e. bounding circle radius) and the creation and update time-stamps in μsec.

Track Estimation-Prediction through Kalman Filtering

Object tracking requires the ability to provide predictions, based on a state space knowledge, which is capitalized from multiple data samplings over time (i.e. observations). This ability resembles the reasoning process that consists in drawing conclusions from facts. A practical solution is attained by formulating this problem as a basic least square state estimation. This approach is, however, limited to a data collection interval (batch method), since, retaining all observations would require unbounded computational and storage resources. Furthermore, least squares formulation does not deal with the inherited uncertainty of measurements and the modeling errors. Indeed, this problem naturally fits in a probabilistic framework.

Addressing the "reasoning" problem from a probabilistic point of view, Kalman [Kal60] proposed a recursive technique that is able to provide accurate state estimates by fusing noisy measurements. This approach is widely referred to as the *Kalman filter* and is principally composed of a two-step closed loop (prediction and update). It

conveniently models the state through the use of two probabilistic moments: its mean (i.e. state vector) and the estimated accuracy, enclosed in its covariance matrix. The Kalman filtering considers the uncertainty as non-biased Gaussian distributions and is constrained to a linear evolution of the states which are perturbed by an additive Gaussian noise. Some particular advantages of the Kalman filtering utilization for the MTT are the adequate response to complex and changing environments, and the easy adaptability to variable sampling frequencies.

In this study, Kalman filtering is used to manage each tracked object independently. The more observations of the same object, the better the predictions about its localization and dynamic (i.e. speed), even if the measurements are provided by a single sensing source (here the ML lidar). This property clearly increases the robustness of the system regarding missed associations, and the management of missed detections and occlusions. It should not be forgotten that the speed of the observed objects is obtained by the means of the Kalman filter.

Hereafter, the Kalman filter basic concepts and equations are briefly presented, assuming that notions of random variables, random process and expectation are clear for the reader. Please consult [TFB00] for a detailed reference.

Kalman Filter Consider the following linear Gaussian system,

$$\begin{cases} \mathbf{x}(t) = \mathbf{A} \cdot \mathbf{x}(t-1) + \boldsymbol{\alpha}(t) \\ \mathbf{z}(t) = \mathbf{C} \cdot \mathbf{x}(t) + \boldsymbol{\beta}(t) \end{cases} \tag{3.4}$$

where $\mathbf{A}$ and $\mathbf{C}$ respectively represent the linear state transition and the observation models. $\mathbf{x}(t-1)$ represents the state vector at the previous sample time. In the same way, $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$ corresponds to the error model and the observation noise. They are supposed to be additive and white zero-mean Gaussian with covariance matrices, $\mathbf{Q}$ and $\mathbf{R}$,

$$\boldsymbol{\alpha}(t) \sim \mathcal{N}\{0, \mathbf{Q}\} \tag{3.5}$$

$$\boldsymbol{\beta}(t) \sim \mathcal{N}\{0, \mathbf{R}\} \tag{3.6}$$

Additionally, the independence between the evolution and the observation error noises is assumed, meaning that they are uncorrelated in the Gaussian case:

$$\mathbb{E}\left[\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t)^T\right] \equiv 0 \tag{3.7}$$

where $\mathbb{E}[\cdot]$ represents the expectation operator. Now, let be $\bar{\mathbf{x}}$ the mean and $\mathbf{P}$ the covariance matrix of the multidimensional system state $\mathbf{x}$:

$$\bar{\mathbf{x}} \doteq \mathbb{E}[\mathbf{x}] \tag{3.8}$$
$$\mathbf{P} \doteq \mathbb{E}\left[(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})^T\right] \tag{3.9}$$

From the above considerations, the Kalman filter consists in determining the system state and its estimated accuracy, from all past observations up to the current time.

In to the following notation, the prediction and update equations of the iterative filtering cycle are presented.

$\mathbf{x}(t\|t)$,	State estimation
$\mathbf{P}(t\|t)$,	Covariance matrix of the estimation error
$\mathbf{z}(t)$	Observation vector
$\mathbf{x}(t\|t-1)$,	Predicted state
$\mathbf{P}(t\|t-1)$,	Covariance matrix of the predicted error

Prediction step

Starting from given initial conditions at $t=0$, the state $\mathbf{x}(0|0)=\mathbf{x}_0$ and the covariance matrix $\mathbf{P}(0|0)$, the system state can be predicted by

$$\mathbf{x}(t|t-1) = \mathbf{A}\cdot\mathbf{x}(t-1|t-1) \tag{3.10}$$
$$\mathbf{P}(t|t-1) = \mathbf{A}\cdot\mathbf{P}(t-1|t-1)\cdot\mathbf{A}^T+\mathbf{Q} \tag{3.11}$$

with

$$\mathbf{x}(t|t-1)) \doteq \mathbb{E}[\mathbf{x}(t)|\mathbf{z}(t-1)] \tag{3.12}$$
$$\mathbf{P}(t|t-1) \doteq \mathbb{E}\left[(\mathbf{x}(t)-\mathbf{x}(t|t-1))\,(\mathbf{x}(t)-\mathbf{x}(t|t-1))^T\right] \tag{3.13}$$

Update step

Once a new observation is available, the predicted state and its covariance can be corrected so as to incorporate the incoming knowledge provided by the state measurements. This knowledge is defined as the *innovation* and is computed through the difference between the measured and the expected state values, $\mathbf{z}(t)-\mathbf{C}\cdot\mathbf{x}(t|t-1)$. Thus,

$$\mathbf{G} = \mathbf{P}(t|t-1)\cdot\mathbf{C}^T\cdot\left(\mathbf{C}\cdot\mathbf{P}(t|t-1)\cdot\mathbf{C}^T+\mathbf{R}\right)^{-1} \tag{3.14}$$
$$\mathbf{x}(t) = \mathbf{x}(t|t-1)+\mathbf{G}\cdot(\mathbf{z}(t)-\mathbf{C}\cdot\mathbf{x}(t|t-1)) \tag{3.15}$$
$$\mathbf{P}(t) = (I-\mathbf{G}\cdot\mathbf{C})\cdot\mathbf{P}(t|t-1) \tag{3.16}$$

where, $\mathbf{G}$ is the optimal *Kalman gain*.

Kinematic Evolution of the Tracked Objects

In order to keep tracking the detected objects, their motion must be predicted by the use of a model. The pertinence of such a model assists the Kalman filter to predict precisely the object dynamic as it moves in space. One can realize a tracking system that classifies the object (e.g. pedestrian, cyclist or vehicle) and then choose an appropriate model for tracking it. Another way can be to track each object by a "batch of filters" running in parallel with different models, and then to choose the better one. This last technique, known as Interacting Multiple Model filtering (IMM) [BBS88], may be quite efficient at the expense of a higher computational cost.

The studied tracking system detects and tracks objects usually present in urban environments, such as pedestrians, cyclists and vehicles. All those objects are observed in a dynamic context not higher to 30 Km/h (i.e.. ego vehicle velocity). Since their observed motions are reported to a fixed reference with a sampling time which is sufficiently higher, they can be assumed to be locally linear with a constant speed during a sample interval, Δt. This model is given as

$$\mathbf{A}_T = \begin{bmatrix} \mathbb{I}_{2\times 2} & \Delta t \cdot \mathbb{I}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbb{I}_{2\times 2} \end{bmatrix} \tag{3.17}$$

Accordingly, the prediction of the track state is given by the following evolution equations:

$$\begin{cases} {}^{\mathcal{W}}\mathbf{y}(t) = \mathbf{A}_T \cdot {}^{\mathcal{W}}\mathbf{y}(t-1) + \boldsymbol{\alpha}_T(t) \\ {}^{\mathcal{W}}\mathbf{y}(t)_{(x,z)} = \mathbf{C}_T \cdot {}^{\mathcal{W}}\mathbf{y}(t) + \boldsymbol{\beta}_T(t) \end{cases} \tag{3.18}$$

with $\mathbf{C}_T = \begin{bmatrix} I_{2\times 2} & \mathbf{0}_{2\times 2} \end{bmatrix}$

where ${}^{\mathcal{W}}\mathbf{y}(t|t-1)$ is the predicted state of the tracked object, ${}^{\mathcal{W}}\mathbf{y}(t)_{(x,z)}$ is the observed track location, $\mathbf{C}_T$ is the observation matrix and Δt is the sampling time period, which is not constant. $\boldsymbol{\alpha}_T(t)$ and $\boldsymbol{\beta}_T(t)$ are additive errors considered as white zero-mean Gaussian noises.

3.2.4 Track-Object Association

After an initial object detection sampling, a new set of tracks is created and all tracks parameters are set up. When a next incoming object arrives (i.e. ML-lidar sampling), iterative tracking actions are performed, starting by a state prediction of all the tracks at the current sampling time. The next intuitive idea, which describes the association process, consists in selecting and assigning the closest new detected object to the predicted track position. This idea leads immediately to the nearest neighbor (NN)

basic concept. Before going straight into the association method, it is important to realize that new objects reports[3] do not represent only objects that have already been observed. But also, a new report could indeed contain new incoming objects in the maneuvering window and spurious detections. The last ones are principally issued of segmentation errors or sensor artifacts. They make complex the object-track assignment.

The simplest association approach is the nearest neighbor. Its naive simplicity makes this method, a "functional approach" under certain conditions. This technique performs object-track association based on a simple track-by-track minimization of an arbitrary distance measure. Indeed, it is not the more robust one, since it has some particular lacks, for instance, in cases when new detected objects lie quite close to existing tracks and in case of missed detections. A more elaborated approach is the Global Nearest Neighbor (GNN), which is based on a object-track assignment matrix. For this, an initial computation of all object-track distance measures is needed. Then, the global solution is found, for instance, as the one which maximizes the number of assignments and minimizes the total cost. This criterion can be generalized by the use of a score function instead of a distance measure.

Gating and Assignment Metrics The NN and GNN methods are both based on object-track distance metrics. For specific applications, object-track assignments could be "evident" occasionally. However, crowded environments are prone to ambiguity, thus, a metric can not be enough discriminant. A *gating* can be used to cope with (or to mitigate) this. It consists in restricting possible assignments to those included in a gate which encloses the predicted track. In case that more than one observation falls within the gate, the "closest" one (i.e. minimal distance metric) is chosen.

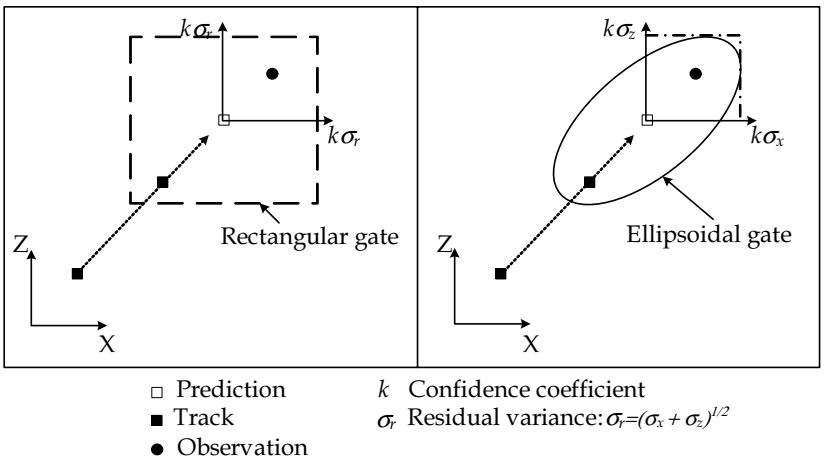


Figure 3.10: *Gating Examples: Different geometrical gate strategies can be used to isolate most feasible observations considered in the object-to-track assignment test.*

[3] A report contains a set of detected objects

Gates are usually characterized by a rectangular or an ellipsoidal region [BP99] (see Fig. 3.10) which is defined in terms of the detected object and the predicted track covariances. Perhaps, the most widely used is the ellipsoidal gate, because it can be efficiently implemented by a simple thresholding of the Mahalanobis distance (also known as χ^2 test) as follows,

$$d_{Mahalanobis}\left({}^{\mathcal{W}}\mathbf{y}(t|t-1),{}^{\mathcal{W}}\tilde{\mathbf{y}}(t)_{(x,z)}\right) = \boldsymbol{\mu}(t)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}(t) \leq Gate \tag{3.19}$$

with

$$\boldsymbol{\Sigma} = \mathbf{M}(t|t-1) + \mathbf{N} \tag{3.20}$$

$$\boldsymbol{\mu}(t) = \mathbf{C}_T \cdot {}^{\mathcal{W}}\mathbf{y}(t|t-1) - {}^{\mathcal{W}}\tilde{\mathbf{y}}(t)_{(x,z)} \tag{3.21}$$

where $d_{Mahalanobis}(\cdot,\cdot)$ represents the Mahalanobis distance operator, $\boldsymbol{\Sigma}$ is the residual covariance matrix, $\boldsymbol{\mu}(t)$ is the state innovation and $\mathbf{M}$ is the covariance matrix of the track. The gate is computed so as to cover a given percentage of possible cases (e.g. 95%) accordingly to a χ^2 distribution.

Alternatively, metrics can also be appropriately chosen, depending on the complexity of the problem. Indeed, some metrics can better reflect (or distinguish) if the predicted track and the observed object are feasibly correlated. In Table 3.1, three different metrics are listed. The Euclidean distance represents only a geometrical indicator about the object-track proximity, which is often not enough discriminant for a MTT problem.

Table 3.1: *Assignment Distance Measurements*

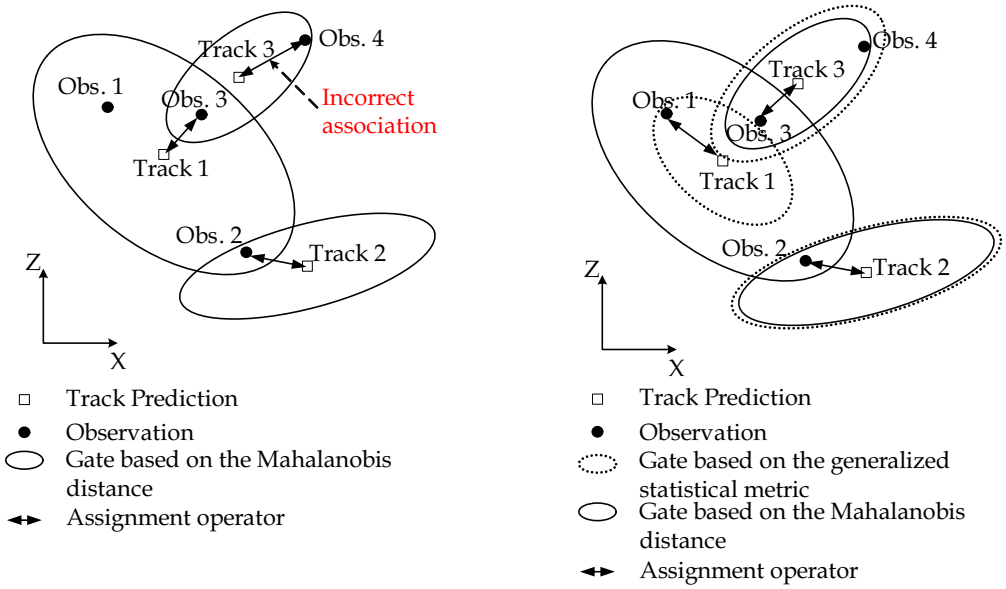
Assignment Distance Measure	Operator	Expression
Euclidean Distance	$d_{Euclidean}(\cdot,\cdot)$	$\|\|\boldsymbol{\mu}(t)\|\|$
Mahalanobis Distance	$d_{Mahalanobis}(\cdot,\cdot)$	$\boldsymbol{\mu}(t)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}(t)$
Generalized Statistical Distance [Bla86]	$d_{Generalized}(\cdot,\cdot)$	$\boldsymbol{\mu}(t)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}(t) + \ln\|\boldsymbol{\Sigma}\|$

A better choice is the Mahalanobis distance which provides a proportional metric criterion based on the uncertainty of the evaluated assignment (log-likelihood measure). Nevertheless, imprecise tracks may still generate large association gates which induces also assignment errors. The generalized statistical distance proposed in [Bla86], penalizes inaccurate tracks by reducing their association gate size as shown in Fig. 3.11b. This penalization is computed as the logarithm of the determinant of the residual covariance matrix as follows,

$$d_{Generalized}\left({}^{\mathcal{W}}\mathbf{y}(t|t-1),{}^{\mathcal{W}}\tilde{\mathbf{y}}(t)_{(x,z)}\right) = \boldsymbol{\mu}(t)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}(t) + \ln|\boldsymbol{\Sigma}| \tag{3.22}$$

where $d_{Generalized}(\cdot,\cdot)$ represents the generalized statistical distance operator. One can

prove that this operator maximizes the object-track association probability.



(a) *Mahalanobis Distance Assignment* **(b)** *Generalized Statistical Distance Assignment*

Figure 3.11: *Association Metric Comparison. In Fig. 3.11a, the observation-to-track assignment using the Mahalanobis distance performs quite well for tracks with reduced uncertainty. It fails, however, in presence of tracks with important uncertainty. The generalized statistical metric deals efficiently with these ambiguous cases as shown in Fig. 3.11b.*

Fig. 3.11 shows an example where three objects are tracked and four objects have been detected. In this example, the object and the track indexes denote implicitly the correct assignment (e.g. Track 1 with Obs 1). The fourth object, here, corresponds to a new observed object in the scene. Hence, it should not be associated to any existing track. In this case, even if the Mahalanobis distance is a quite good statistical association metric, it can sometimes fail in presence of tracks with important uncertainty, as observed in the assignment of Track 1 with the observation, Obs. 3 (see Fig. 3.11a). In contrast, the use of the generalized statistical metric deals efficiently with this ambiguous case penalizing the association of inaccurate tracks. A visual interpretation of using the generalized distance is that it decreases the size of ellipsoidal gate as shown in Fig. 3.11b particularly for Track 1.

Managing Ambiguity As presented hereinbefore, nearest neighbor methods constitute mono-hypothesis association approaches. These methods address ambiguous assignment cases by retaining the best available observation (i.e. the closest one). More sophisticated methods address this issue like the Joint Probabilistic Data Association (JPDA) which is an extension, for the MTT problem, of the Probabilistic Data Association (PDA) in Single Target Tracking (STT) [BP99]. JPDA deals with multiple observations included in a track gate, by combining "all-neighbors" observations. Such

a combination takes into account single object-track and joint assignment uncertainty. The use of this technique in a filtering process is known as JPDAF.

Extending the JPDA concept, multiple hypothesis can be created instead of combining them. Following this scheme, ambiguities can be solved in the measure that detection reports provides enough "evidence" to determine which assignment is correct or not. In other words, the algorithm defers logically the assignment decision. This approach is called Multiple Hypothesis Tracking (MHT) and is considered sub-optimal, since the number of hypotheses must be continuously controlled to avoid an exponential proliferation. Thus, redundant hypothesis branches must be merged and unlikely ones must be pruned.

3.2.5 Closing the tracking loop

After a brief survey of a few assignment approaches related to MTT, we have limited the scope of this study to a simple assignment method able to provide acceptable results within the maneuvering window in urban conditions. Thus, a mono-hypothesis NN approach based on the generalized statistical metric with an ellipsoidal gating constraint is considered for further processes.

Consequently, assigned tracks states and their corresponding covariances are updated through the Kalman filter update step equations. In such a way, a fusion of the associated lidar objects and the predicted tracks is done.

The tracking cycle is not yet finished. There are still some important issues to manage: Track initiation and deletion. The initiation step considers new appearing objects in the maneuvering window as those which have not been assigned to a track. Hence, new tracks are created from their measured location. In theory, this function is able to initialize as many tracks as unmatched objects. However, embedded applications imply that computational resources are limited. This is why, a fixed number of tracks is set, to track a reasonable quantity of objects.

On the other hand, the deletion mechanism is necessary to determine, under given conditions, which tracks should be conserved or deleted from the track list. In a NN tracking approach, deletion is usually limited to deal with non-associated tracks only. Defining a stable criterion to delete tracks can be tricky, since the global tracking performance can be adversely affected. A simple thresholding can be useful, based on the time or the numbers of reports since the last update stamp. In this way, the tracking algorithm can cope with object occlusions. However, setting a long prediction time may lead to keeping track artifacts.

3.3 Visual Confirmation

The presented perception system is able to provide a precise 3D localization, with respect to a fixed-reference frame of the ego vehicle and of the surrounding objects that could represent potential obstacles (i.e. withing the maneuvering window). The considered tracking functionality does not intend to recognize or classify the detected objects, but only to track them in space. As depicted in Fig. 3.12, the object tracking is obtained through the interaction of two complementary sensing processes: Ego-localization and object detection.

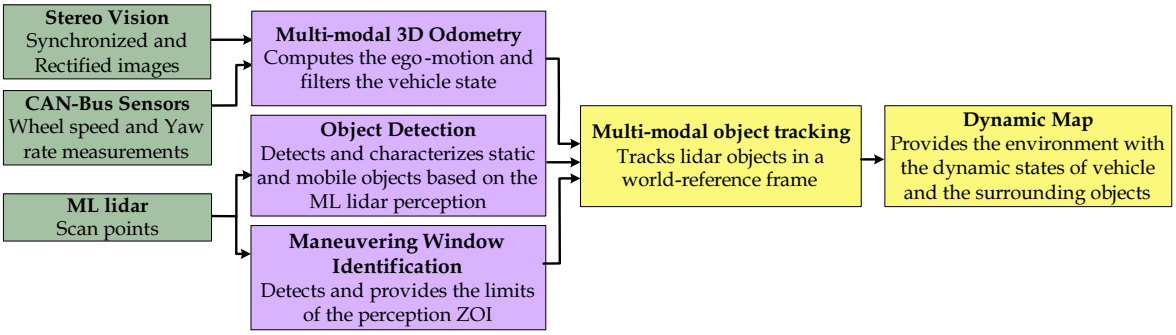


Figure 3.12: *Perception functions and their interactions for the object localization and tracking*

The proposed system can provide valuable information to enhance functional safety for passenger cars, such as collision warning systems. These envisioned applications are focused on Advanced Driver Assistance Systems, which intends to increase car safety by helping users (driver) in the driving process. In this context, the driver keeps complete control of the vehicle and is only assisted through warnings.

Previous works on ergonomics of alarm design have studied how the reliability of warning signals impacts driver performance. Bliss and Acton [BA03] reveals that False Alarms (for short FAs) are potentially more harmful than non-detections. This is because FAs have a huge potential to induce driver distraction, causing users to redirect their attention away from the primary driving task, as demonstrated in [Bab94]. Indeed, a common reaction of the driver would be to switch the system off, if it issues too many FAs.

In the considered system, the object detection relies on a single source: The ML lidar. This source is quite efficient to detect and localize obstacles. Unfortunately, alarms can arise from non-hazardous objects such as those located on the sidewalk. For this, the spatial filtering of the detected objects is done with the use of the maneuvering window, tracking in this way objects situated within the road only. However, sometimes spatial filtering is not enough, since FAs can still be issued from segmentation errors or spurious detections, also called, phantom measurements (due to snowflakes and heavy rain for instance). A second independent sensing source can then be used to confirm

the existence of a real object. In such a case, warnings emitted by both detection sources would be very confusing for the driver. Warnings must be reliable and clear, given the short time the driver has to analyze and to perform appropriate actions. Thus, the integrity of the information provided by the warning system is essential for driver trust.

A first effort to strengthen the integrity of the multi-modal system output was done by integrating over time the succeeding object detection reports (temporal filtering). A complementary way is to use different perception methods checking that same causes produce same effects. Integrity is one of the major attributes related to the performance of ADAS functions. To ensure this, information has to be sensed by at least two different sensor principles [Sti01]. If the integrity is checked with respect to a given level of confidence, appropriate warnings can be emitted by the ADAS.

The strategy proposed here consists in a visual confirmation method of the detected and localized targets using lidar as summarized in Fig. 3.13. This approach does not treat missed detections of the ML-lidar. On the contrary, it filters FAs and, thus, increases integrity of warning messages.

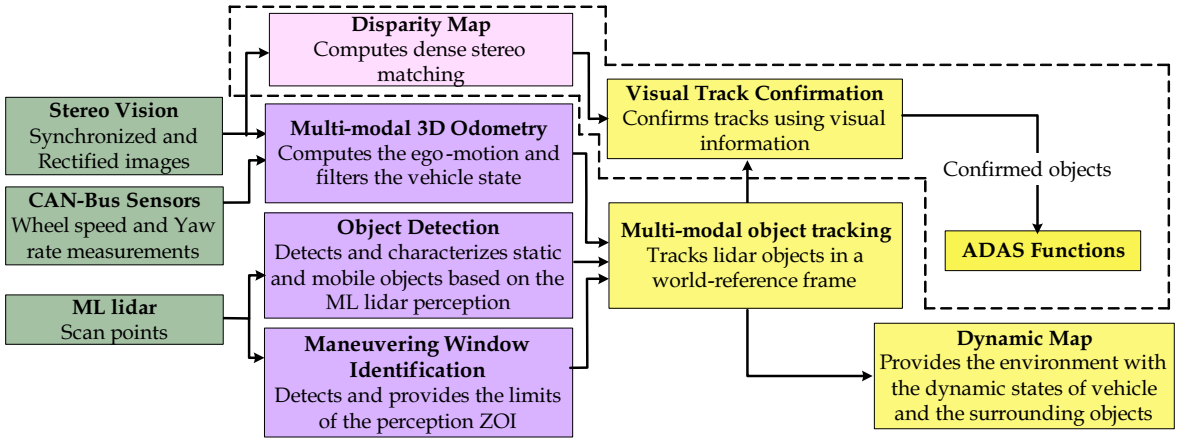


Figure 3.13: *Multi-modal object localization and tracking with a visual track confirmation module (dashed line section) intended to strengthen the integrity of the output system.*

Visual track confirmation is performed using the following strategy. Firstly, each lidar-tracked object is transformed into a ego vehicle frame (denoted $\mathcal{E}(t)$). Its corresponding bounding cylinder (lidar bounding circle with an arbitrary height) is reprojected onto the stereo images. In each image, the track projection provides a Region Of Interest (ROI). Secondly, the pixels composing the ROI are reconstructed by stereopsis[4] in the 3D space in order to obtain a dense 3D point cloud. Afterward, this set of 3D points is segmented into 2 clusters: The object and the background. At last, the track is confirmed if one of the 3D points cluster centroids is assigned to the lidar object using a Mahalanobis distance given a confidence level. Indeed, FAs which do not validate the χ^2 test are filtered.

[4]Reconstruction of the 3D space as a result of binocular disparity.

3.3.1 Frames used in solving the visual confirmation problem

The process of confirming tracks through visual information requires the transformation of geometrical data between different frames. The SVS frame, located in the baseline midpoint of the cameras, is oriented forward and slightly down-angled towards the road. This orientation is desirable for visual odometry, since it reduces poor textured image regions induced by the sky (see Fig. 3.14). Nevertheless, this configuration makes more complicated the geometrical scene analysis.

Figure 3.14: *Image samples showing that the SVS orientation is tilted on the road*

The ego frame $\mathcal{E}(t)\{Right\text{-}Down\text{-}Front\}$ is then defined so as to have its ${}^{\mathcal{E}}XZ$ plane parallel to the road surface, which greatly simplifies the problem. The knowledge of the vehicle pitch angle w.r.t. the road plane can provide a dynamic attitude correction of this frame. In this study, however, the orientation of the ego frame is initially chosen and no further correction is applied.

The mobile perception makes use of three principal frames: World, Ego (i.e. body frame) and SVS, respectively denoted as $\mathcal{W}$, $\mathcal{E}(t)$, and $\mathcal{S}(t)$. As it can be remarked by their time-dependency, the ego and SVS frames are mobile references, rigidly linked to the vehicle.As it is illustrated in Fig. 3.15, ego and SVS frames are coincident in position (i.e. midpoint of the SVS cameras) but differ in attitude.

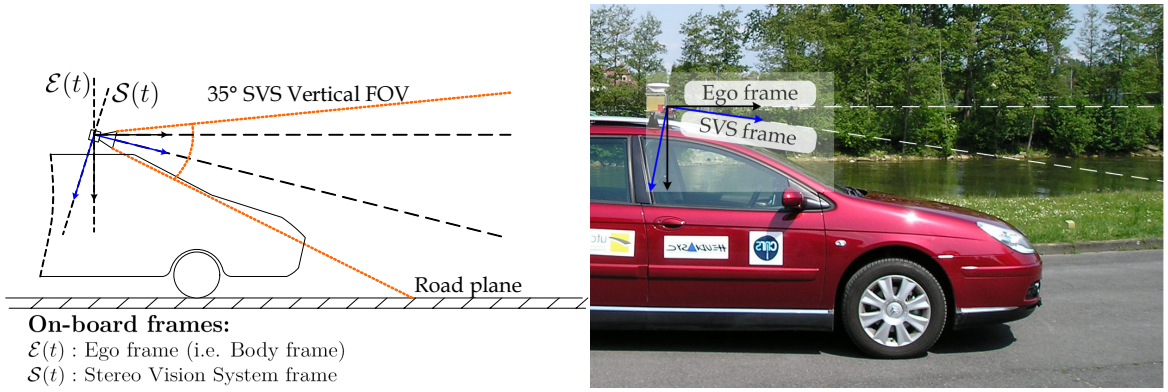


Figure 3.15: *On board frames. The ego frame $\mathcal{E}(t)\{Right\text{-}Down\text{-}Front\}$ attitude is such that the ${}^{\mathcal{E}}XZ$ plane is parallel to the road surface, which simplifies the geometrical information analysis. Its position is coincident with the SVS frame.*

For practical reasons, the world frame is selected coincident in attitude and position as the ego frame at initial time $t = 0$. These assumptions remain appropriate to the cooperative perception approaches [SAF], since the world frame can be easily reported to an ENU Geo-localized frame (as detailed in section 2.7.2 of chapter 2).

An illustration of the evolution of the frames at three sampling times is depicted in Fig. 3.16.

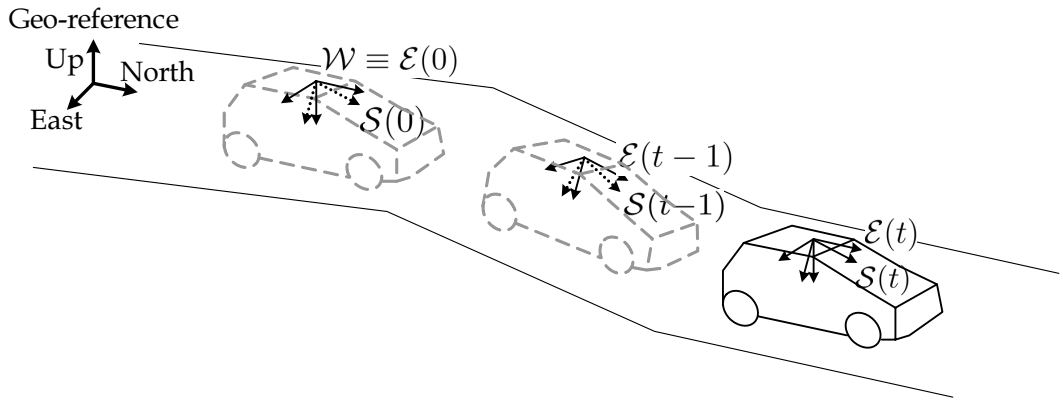


Figure 3.16: *Frame evolution over three sample times. The ego frame is initially chosen so as to be oriented parallel to the road plane. Then, it evolves in space rigidly linked to the SVS frame. The world frame is selected as the ego frame at time $t = 0$.*

3.3.2 Region Of Interest in the Images

Because we are interested in a 3D dense reconstruction in a limited area, the ROIs extraction is an efficient way to reduce complexity. Let us recall that the object tracking is performed in the world frame: Tracks positions must be referenced w.r.t. a camera frame, in order to be projected onto images. Before doing that, the 3D track position must be reconstructed, since the state vector contains only its planar coordinates (i.e. ${}^{\mathcal{W}}XZ$ coordinates). For this, the last object altitude (i.e. ${}^{\mathcal{W}}Y$ coordinate) associ-

ated to the track is used. Hereafter, the reconstructed 3D track fix, in the world frame, is denoted by ${}^{\mathcal{W}}\mathbf{y}^{+}(t|t-1) = {}^{\mathcal{W}}[\mathbf{y}_x, \mathbf{y}_y, \mathbf{y}_z]^T$ where $\mathbf{y}_y$ is the added altitude coordinate.

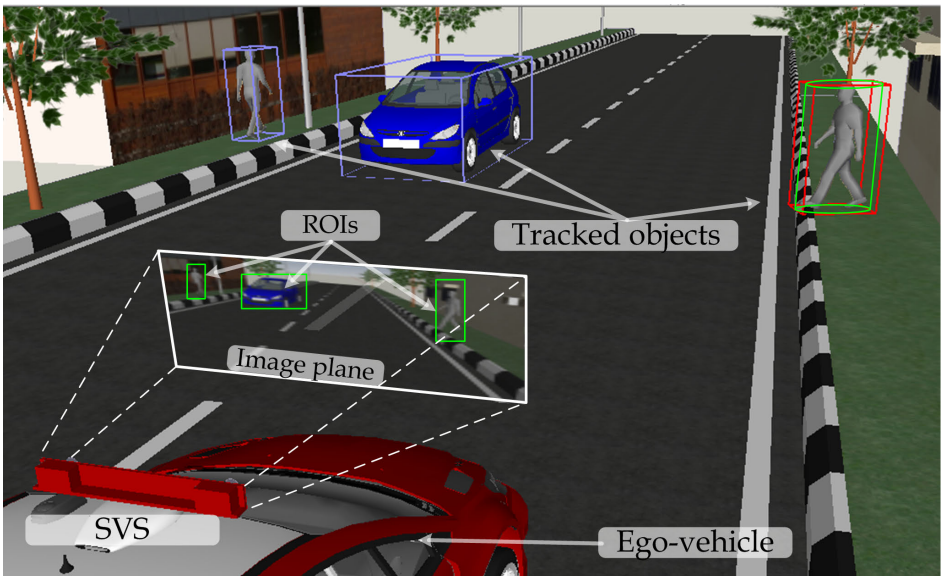


Figure 3.17: *Estimation of image ROIs through the re-projection onto the image plane of the track bounding volume. Four contour points of the track bounding volume are determined from the track 3D fix and the radius of its bounding cylinder. The projection of such points onto images characterizes the track ROI.*

We determine four contour points of the track bounding volume using the 3D coordinates of its center and the radius of its bounding cylinder. The projection of such points onto images characterizes the ROI, as depicted by Fig. 3.17.

The reconstruction of the bounding volume supposes that observed objects are orthogonal to the road plane. This assumption can be easily applied in $\mathcal{E}(t)$, since the ${}^{\mathcal{E}}XZ$ plane is parallel to this surface. Consequently, tracks are localized in the ego frame, $\mathcal{E}(t)$, through the following rigid transformations:

$$
\begin{aligned}
{}^{\mathcal{S}(t)}\underline{\mathbf{y}}^{+}(t|t-1) &= {}^{\mathcal{W}}\bar{\mathbf{q}}_{\mathcal{S}(t)} \star \left({}^{\mathcal{W}}\underline{\mathbf{y}}^{+}(t|t-1) - {}^{\mathcal{W}}\underline{\mathbf{t}}_{\mathcal{S}(t)} \right) \star {}^{\mathcal{W}}\mathbf{q}_{\mathcal{S}(t)} && (3.23)\\
{}^{\mathcal{E}(t)}\underline{\mathbf{y}}^{+}(t|t-1) &= {}^{\mathcal{E}}\mathbf{q}_{\mathcal{S}} \star {}^{\mathcal{S}(t)}\underline{\mathbf{y}}^{+}(t|t-1) \star {}^{\mathcal{E}}\bar{\mathbf{q}}_{\mathcal{S}} && (3.24)
\end{aligned}
$$

with

$$
\begin{aligned}
{}^{\mathcal{W}}\mathbf{q}_{\mathcal{S}(t)} &= {}^{\mathcal{E}}\mathbf{q}_{\mathcal{S}} \star {}^{\mathcal{S}}\mathbf{q}_{\mathcal{S}(t)} && (3.25)\\
{}^{\mathcal{W}}\underline{\mathbf{t}}_{\mathcal{S}(t)} &= {}^{\mathcal{E}}\mathbf{q}_{\mathcal{S}} \star {}^{\mathcal{S}}\underline{\mathbf{t}}_{\mathcal{S}(t)} \star {}^{\mathcal{E}}\bar{\mathbf{q}}_{\mathcal{S}} && (3.26)
\end{aligned}
$$

where ${}^{\mathcal{S}(t)}\mathbf{y}^{+}(t|t-1)$ is the 3D track coordinates in the $\mathcal{S}(t)$ frame, ${}^{\mathcal{E}(t)}\mathbf{y}^{+}(t|t-1)$ is the resulting position of the track in $\mathcal{E}(t)$, ${}^{\mathcal{W}}[\mathbf{q},\ \mathbf{t}]_{\mathcal{S}(t)}$ is the rigid transformation from $\mathcal{S}(t)$ to $\mathcal{W}$ and ${}^{\mathcal{E}}\mathbf{q}_{\mathcal{S}}$ is the attitude between the SVS and the ego mobile frames. The transformation ${}^{\mathcal{S}}[\mathbf{q},\ \mathbf{t}]_{\mathcal{S}(t)}$ corresponds to the 3D vehicle motion estimated in Eq. 2.70

and 2.71. Hereinbefore, the frame indexes corresponding to $t = 0$ have been conveniently omitted. Eq. 3.23 performs an inverse transfer. This is why ${}^{\mathcal{W}}[\mathbf{q},\, \mathbf{t}]_{\mathcal{S}(t)}$ has been used to obtain the track position in $\mathcal{S}(t)$.

The contour points coordinates, ${}^{\mathcal{E}(t)}\mathbf{p}_{\{1,\ldots,4\}}$, are computed as follows,

$$
\begin{aligned}
{}^{\mathcal{E}(t)}\mathbf{p}_1 &= {}^{\mathcal{E}(t)}\mathbf{y}^{+}(t|t-1) + [r,\, h_{min},\, 0]^T && (3.27)\\
{}^{\mathcal{E}(t)}\mathbf{p}_2 &= {}^{\mathcal{E}(t)}\mathbf{y}^{+}(t|t-1) + [-r,\, h_{min},\, 0]^T && (3.28)\\
{}^{\mathcal{E}(t)}\mathbf{p}_3 &= {}^{\mathcal{E}(t)}\mathbf{y}^{+}(t|t-1) + [-r,\, -h_{max},\, 0]^T && (3.29)\\
{}^{\mathcal{E}(t)}\mathbf{p}_4 &= {}^{\mathcal{E}(t)}\mathbf{y}^{+}(t|t-1) + [r,\, -h_{max},\, 0]^T && (3.30)
\end{aligned}
$$

where r is the radius of its bounding cylinder and $[h_{min},\, h_{max}]$ is an interval denoting the track height (see Fig. 3.18).

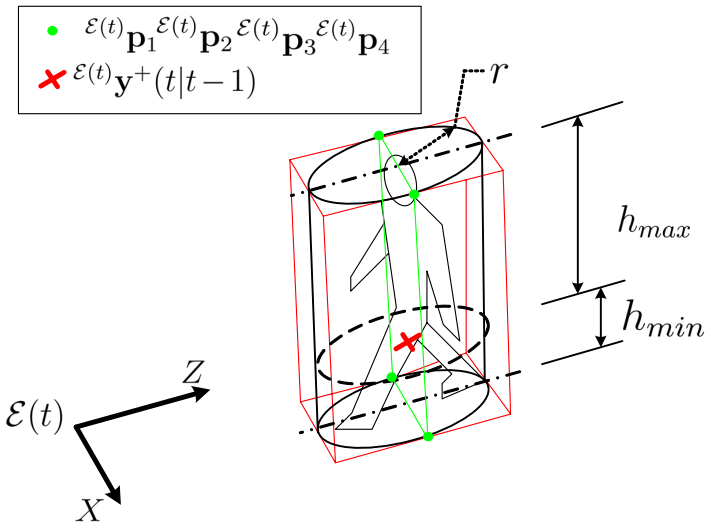


Figure 3.18: *Computation of the contour points of the track.*

Finally, the ROI coordinates, in the left camera image of the SVS, are determined by projecting of points ${}^{\mathcal{E}(t)}\mathbf{p}_{\{1,\ldots,4\}}$ onto the image plane using the following equation:

$$
\begin{bmatrix} u \\ \mathrm{v} \\ 1 \end{bmatrix} \sim \mathbf{K} \cdot \left(\left({}^{\mathcal{S}}\mathbf{q}_{\mathcal{E}} \star^{\mathcal{E}(t)} \underline{\mathbf{p}}_{\{1,\ldots,4\}} \star^{\mathcal{S}} \bar{\mathbf{q}}_{\mathcal{E}} \right) +^{\mathcal{C}} \underline{\mathbf{t}}_{\mathcal{S}} \right) \qquad (3.31)
$$

where $[u\ v\ 1]^T$ are the subpixelic image coordinates and $\mathbf{K}$ is the left camera intrinsic matrix. ${}^{\mathcal{S}}\mathbf{q}_{\mathcal{E}}$ is the rotation which transfers the point coordinates from $\mathcal{E}(t)$ to $\mathcal{S}(t)$ and ${}^{\mathcal{C}}\mathbf{t}_{\mathcal{S}} = -{}^{\mathcal{S}}\mathbf{t}_{\mathcal{C}}$ is the translation which locates the points from $\mathcal{S}(t)$ into $\mathcal{C}$, the left camera frame. The operator $\sim$ means up to a scale factor.

We have chosen to precompute a dense disparity map of the complete image, since it is suitable when too many ROIs have to be confirmed. As nowadays many robotics

cameras are able to deliver dense disparity maps and stereo images simultaneously, it constitutes an easy-to-access information.

3.3.3 3D Dense Reconstruction of the ROI

Every pair of images contains 3D dense information of the observed scene, since the pixels stereo correspondence and the intrinsic camera parameters are known. As the estimated track position has been projected onto the image, a limited 3D reconstruction is necessary.

Let us consider that the pixel stereo correspondence is represented by a dense disparity map referenced w.r.t. the left camera of the SVS. This representation consists in a bi-dimensional mapping, denoted $\mathbf{D}_d$, where precomputed disparity values are arranged. Such values are the differences, in x-coordinates on the image planes, of the same features viewed in both cameras as follows:

$$\mathbf{D}_d(u,\, v) = u - u' \tag{3.32}$$

where (u, v) and (u', v') are the pixel coordinates of two corresponding points. The disparity is denoted d.

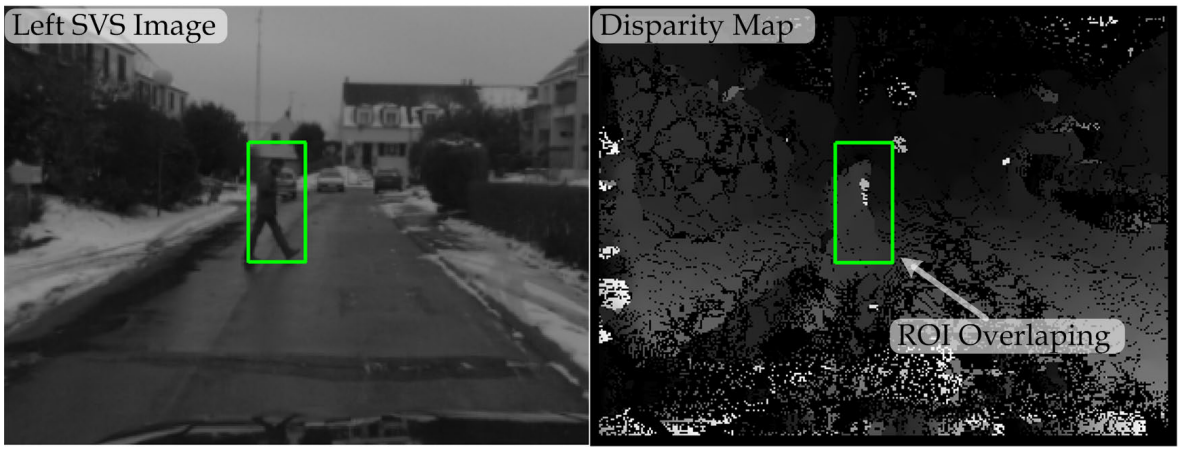


Figure 3.19: *Overlap of the estimated image ROI onto the disparity map. Depth information contained within the overlapped ROI is used to perform 3D dense reconstruction.*

The proposed 3D dense reconstruction consists firstly in overlapping the ROI onto the disparity map as illustrated in Fig. 3.19. The 3D metric coordinates ${}^{\mathcal{C}}[x\, y\, z]^T$ of each pixel included in the ROI are estimated by performing a classical triangulation process [HZ03]. This process assumes a calibrated SVS with rectified stereo images, which under ideal conditions can be modeled as illustrated in Fig. 3.20.

The depth coordinate of each point can be derived straightforwardly thanks to the

relation of similar triangles:

$$\begin{aligned} \frac{b-u+u'}{{}^{c}z-f} &= \frac{b}{{}^{c}z} \\ {}^{c}z &= \frac{fb}{u-u'} \end{aligned} \tag{3.33}$$

The x and the y coordinates are determined using the same principle:

$${}^{c}x = \frac{u\,{}^{c}z}{f} \tag{3.34}$$

$${}^{c}y = \frac{v\,{}^{c}z}{f} \tag{3.35}$$

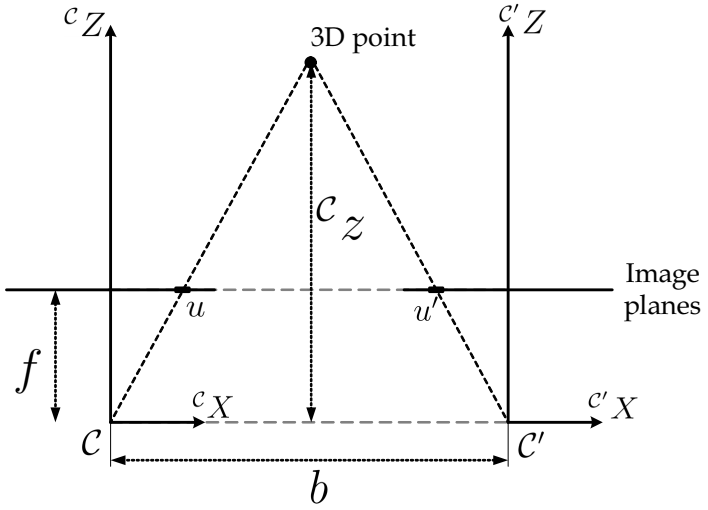


Figure 3.20: *Stereo triangulation model for rectified stereo images*

3.3.4 Track Confirmation Test

In our study, the visual confirmation is performed using an occupancy space analysis. Conceptually, the existence of a tracked object is confirmed if its position matches with the 3D visual points.

One possible strategy is to test all the reconstructed points in order to determine the percentage that matches the track location. A previous work [PLR+06] has already examined a stereo vision confirmation method of tracks, for a long-range obstacle detection system, based on a single-raw lidar. To this end, visual information is analyzed in a parameter space representation known as v-disparity[5]. Re-projecting the ROI provided by the lidar in this parameter space, obstacles can then be confirmed according to three different proposed criteria: the total number of obstacle-pixels, the prevailing

[5]Parameter space representation where the $v-$image coordinate and the corresponding disparity value of a pixel determine its plot location.

alignment orientation or the lidar points attitude. The track confirmation scheme is conducted following a different approach w.r.t the one considered by Perrollaz et al. Our technique might be similar to the criterion based on the number of obstacle-pixels proposed in [PLR+06], but performed in the 3D space. However, such an approach is time consuming and less suitable for an embedded real-time system.

An alternative is to cluster the points reconstructed by vision in order to handle object-like representations. Track integrity can then be assessed by testing if the lidar track is highly correlated with the object-like clusters. This approach can avoid many calculations and the cluster uncertainties can be integrated into a verification test. Let us study how it works in the following.

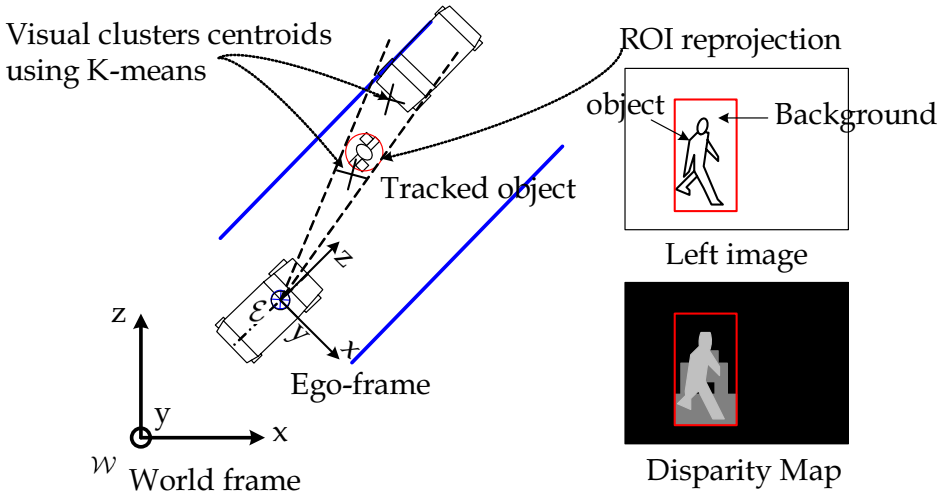


Figure 3.21: *Visual 3D Track Confirmation. Track integrity can then be assessed by simply testing if the lidar track is highly correlated with the object-like clusters issued of the visual ROI dense 3D reconstruction.*

Looking at the lidar cylindrical object model, it can be noticed that image ROI contains most of the time, not only the tracked object, but also, some scene background regions, as illustrated by Fig. 3.19. This phenomenon can be commonly evidenced in the space distribution of the dense ROI reconstruction. For these reasons, the clustering of the 3D points can be seen as an ideal solution for simplifying and speeding up the track verification process.

The clustering-based track verification approach is applicable assuming that the objects and the scene background within the ROI are distinguishable in the 3D space (see the example given in Fig. 3.21). If this assumption holds, the reconstructed 3D points can be segmented into two object-like clusters: track and background. This assumption is usually justifiable, because the tested objects are located on the maneuvering window which corresponds to the navigable space, as shown in Fig. 3.22, for a real case example.

Here, a k-Means method [Mac67] is particularly suitable, since the number of classes is known. Besides, this clustering heuristic is efficient and can converge rapidly.

The clustering algorithm consists in an iterative routine composed of two mere steps: the expectation and the maximization. In the expectation step, each reconstructed 3D point is assigned to the cluster whose centroid[6] is the closest, according to a distance criterion. The maximization step consists in updating the cluster centroid where the new point was assigned to.

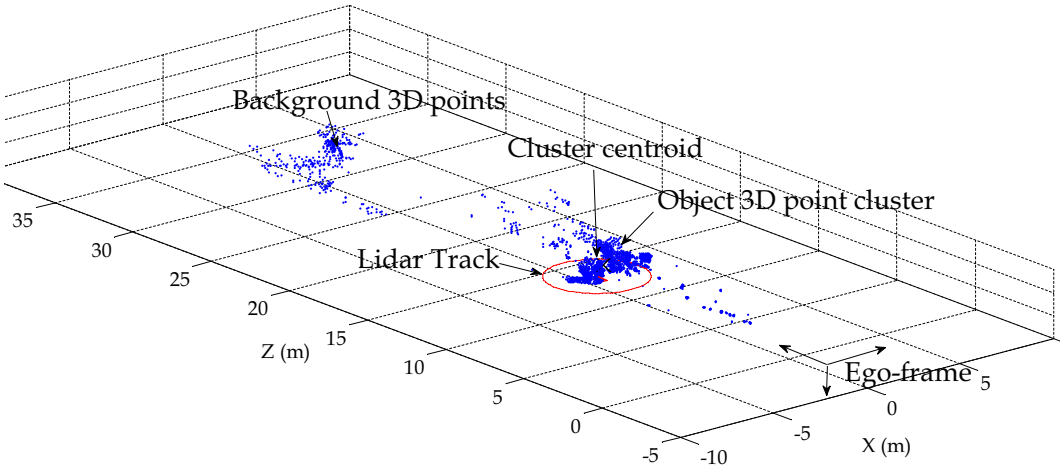


Figure 3.22: *Visualization in space of a confirmed track using 3D point clusters (Real data).*

The iterative cycle is repeated until the cluster point assignment does not change any more. Hence, the determined centroid clusters are considered stables. The algorithm provides then 2 clusters characterized by theirs centroids, ${}^{\mathcal{E}(t)}\mathbf{c}_j$ for all $j = \{1, 2\}$, and their 3D associated points.

For our practical problem, k-Means works quite well. However, it must be highlighted that the theoretical convergence into the global optimum of this clustering method is no guaranteed. To mitigate this, it is common to repeat the clustering process with different initial conditions and take the best result (trade-off solution).

Once the clusters have been determined, the uncertainty associated with every cluster needs to be estimated. Representing the location uncertainty of the clusters requires a particular attention. For this purpose, the "*most expected*" position of every cluster has been considered to be the average position of the points composing it. The idea is to model the cluster location uncertainty by a covariance matrix, estimated with a confidence score. As the centroid is determined from a set of reconstructed points, its confidence score is indeed strongly correlated with the triangulation process quality.

As shown in [BH87], the uncertainty in range (${}^{\mathcal{S}(t)}Z$) and horizontal (${}^{\mathcal{S}(t)}X$) position depends exclusively on the horizontal image coordinates disparity of a stereo associated feature. Their statistical independence from the vertical image coordinate are explicitly formulated by the triangulation equations (Eq. 3.34, 3.35, 3.33).

In [BH87], Blostein and Huang have examined firstly a probabilistic interpretation of the geometrical point projection process onto stereo image planes. Then, they have

[6]The centroid corresponds to the mean point of the cluster.

determined the joint probability distribution of a 3D point within a Region Of Uncertainty (ROU), reprojected on the left and right image planes. This joint probability distribution is directly used to derive the uncertainty in range of a point using triangulation.

Accordingly, the integration of the joint probability J_p of the relative error in range, constrained to be within a certain tolerance α is given by:

$$J_p = \begin{cases} \frac{1}{\ln(1-d^{-2})}\left(\frac{2\alpha(\alpha-d^{-1})}{1-\alpha^2} + \ln(1-\alpha^2)\right) & \alpha < \frac{1}{d} \\ 1 & \alpha \geq \frac{1}{d} \end{cases} \tag{3.36}$$

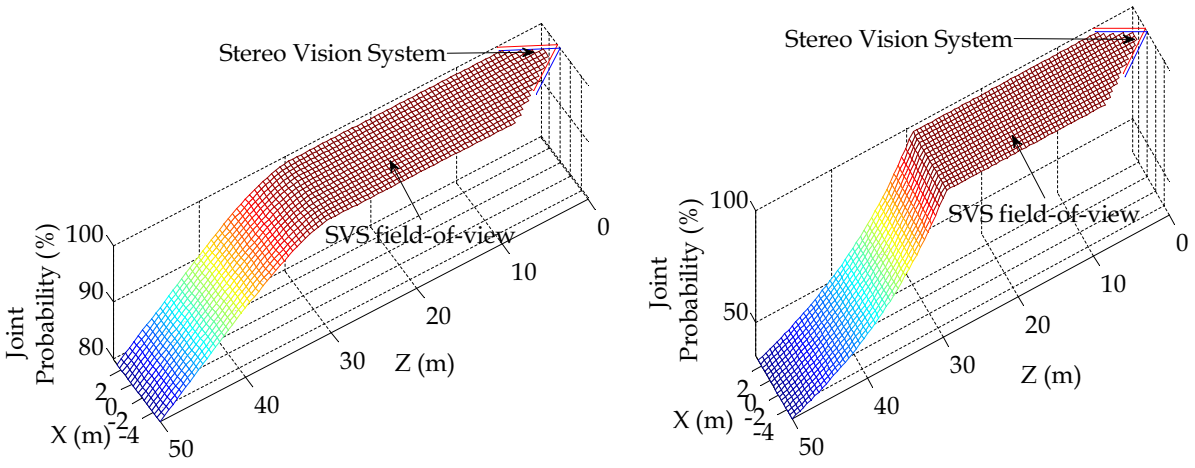
where we recall that d denotes the disparity.

The complexity of this expression can be significantly simplified using a first order Taylor series approximation of the natural logarithm terms. Hence, the approximate expression is reduced to the following:

$$J_p = \begin{cases} 1 - \left(1 - (\alpha \cdot d)^2\right) & , \alpha < \frac{1}{d} \\ 1 & , \alpha \geq \frac{1}{d} \end{cases} \tag{3.37}$$

Accurate and approximated error models within the field of view of the stereo vision system are illustrated respectively in Fig. 3.23a and 3.23b.

This error function assesses the SVS capabilities for estimating a range measurement with a given accuracy. The larger the disparity of the imaged object, the better the estimates of its 3D location. Logically, if an object is located at a distance where its disparity in less than a pixel, its location estimate will be quite inaccurate.



(a) *Accurate Triangulation Error Model* **(b)** *Approximated Triangulation Error Model*

Figure 3.23: *Error model behavior in the SVS Field-of-View*

We adapt this error model for estimating a confidence score, denoted τ_j. The score τ_j expresses the uncertainty in the cluster centroid fix, ${}^{\mathcal{E}(t)}\mathbf{c}_j$, in depth w.r.t the SVS.

Thus, τ_j constrained to a given tolerance factor α, is given as follows,

$$\tau_j = \begin{cases} 1 - \left(1 - \left(\alpha \cdot \frac{b \cdot f}{{}^{\mathcal{E}(t)}\mathbf{c}_{(z),j}}\right)^2\right) & , \alpha < \frac{{}^{\mathcal{E}(t)}\mathbf{c}_{(z),j}}{b \cdot f} \\ 1 & , \alpha \geq \frac{{}^{\mathcal{E}(t)}\mathbf{c}_{(z),j}}{b \cdot f} \end{cases} \tag{3.38}$$

where ${}^{\mathcal{E}(t)}\mathbf{c}_{(z),j}$ means, with an abuse of the notation, the Z metric coordinate of the centroid ${}^{\mathcal{E}(t)}\mathbf{c}_j$. The tolerance error factor, α, is set in meters, taking into account the image resolution, the focal distance, the baseline of the SVS and the accuracy of the SVS calibration.

In order to perform a statistical assignment test between the clusters and the candidate track, a suitable estimate for the covariance matrix of the 2D position (i.e. in the plane ${}^{\mathcal{E}(t)}XZ$) of the cluster centroid $\mathbf{U}$ can be estimated by making use of the confidence score τ_j:

$$\mathbf{U} = \begin{bmatrix} \frac{1}{k_1 \cdot \tau_j} & 0 \\ 0 & \frac{1}{k_2 \cdot \tau_j} \end{bmatrix} \tag{3.39}$$

where the choice of the weighting parameters k_1 and k_2 can be made on the basis that the reconstruction error regarding to the depth has more impact in the longitudinal direction(${}^{\mathcal{E}(t)}Z$ axis) than transversely (i.e. $k_2 > k_1$)[7].

Each cluster is tested using a Mahalanobis distance w.r.t the ML-lidar tracked object position:

$$d_{Mahalanobis}\left({}^{\mathcal{E}(t)}\mathbf{c}_{(x,z),j}, {}^{\mathcal{E}(t)}\mathbf{y}(t|t-1)_{(x,z)}\right) = \kappa \cdot (\mathbf{U} + \mathbf{M}(t|t-1))^{-1} \cdot \kappa^T \tag{3.40}$$

with $\kappa = {}^{\mathcal{E}(t)}\mathbf{c}_{(x,z),j} - {}^{\mathcal{E}(t)}\mathbf{y}(t|t-1)_{(x,z)}$

where ${}^{\mathcal{E}(t)}\mathbf{c}_{(x,z),j}$ and ${}^{\mathcal{E}(t)}\mathbf{y}(t|t-1)_{(x,z)}$ are respectively the ${}^{\mathcal{E}(t)}XZ$ coordinates of the centroid cluster to be tested and the tracked object. The matrix $\mathbf{M}(t|t-1)$ is the covariance of the tracked object location.

If a cluster is located within the ellipsoidal gate, the system can establish that the visual information source matches the lidar track position. If the two clusters fall within the gate, the closest one is chosen. A typical value for the gating threshold corresponds to 3 standard deviations[8] in order to compensate approximations due to track motion and imprecisions in case of important pitch vehicle changes. The success of the test confirms the real existence of the tracked object. Thus, the integrity of this object is enhanced, since two independent sensing sources have been used.

If the assignment test fails, the system does not destroy the object track, it remains to

[7] typically one can choose $k_2 = 2k_1$

[8] Confidence interval of 99% for 2 DOF, according to Chi-square distribution (χ^2)

be a feasible object hypothesis. However, its existence evidence is not enough trustworthy to be provided as an outcome.

So far, we have considered that the ROI contains only the object and background regions. Two particular cases can sometimes happen when the estimated track location is re-projected onto the image plane.

The first case occurs when the ROI contains only the object (i.e. absence of background). In this case, the track confirmation will still provide a correct result, because the two clusters will remain close to each other and close to the real track.

The second possible case arises when no real object is included in the ROI (i.e. a lidar false detection). Thus, the two estimated clusters will make part of the background and the track will not be confirmed.

So, we can conclude that these two worst-cases have no direct impact in the clustering convergence and visual confirmation.

3.4 Multi-rate information management

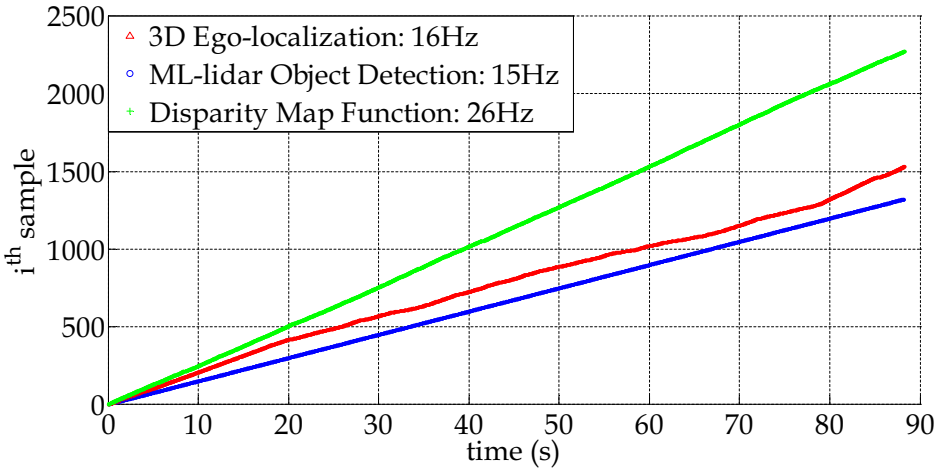
With the aim of increasing safety during driving, ADAS functions require accurate, pertinent and reliable information about the vehicle's surrounding. Based on this information, a suitable scene understanding level with an adequate time response according to the dynamic context has to be provided. Single sensor based techniques are usually not good enough, since the observed environment is often complex. Each sensor has a limited perception of the total space representation. Coping with sensor limitations entails the use of multiple sensing strategies. In this way, multi-modal systems achieve complementary field-of-view sensing and redundant perception of important information. One can imagine, then, a fused environment representation issued of different specialized perception functions, and providing pertinent information to high-level ADAS safety functions (e.g. pedestrian detection and collision warning).

To this end, multi-sensor data must be synchronized ensuring the time consistency of the system. Besides, the use of synchronized information simplifies the association data process.

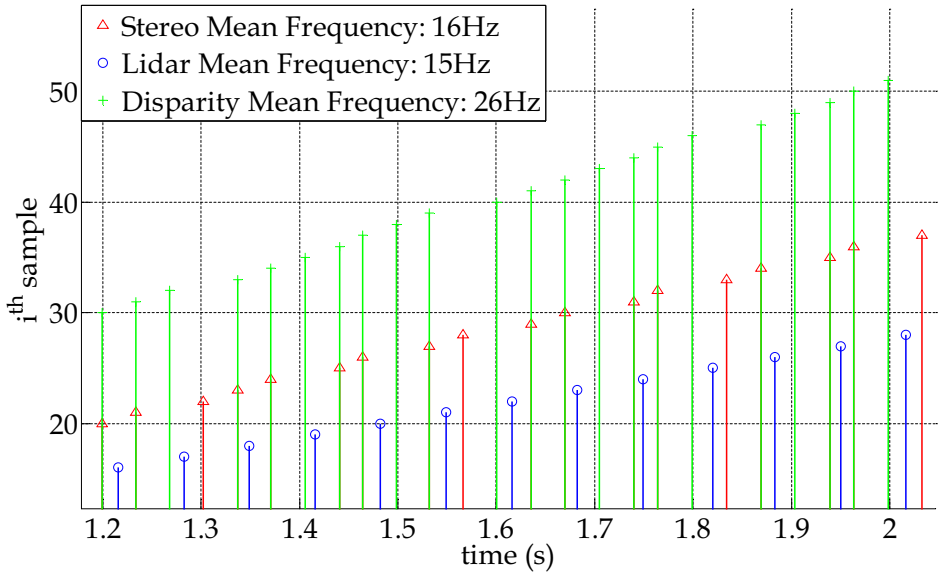
In our application synchronization is crucial, since visual track confirmation requires that image information, vehicle localization and lidar reports correspond exactly to the same instant. Indeed, the fusion module architecture relies on three main functions interacting together: Disparity map computation, 3D-ego localization and object detection. These functions are asynchronous and run in different threads at different frequencies (the mean frequencies are respectively: 26, 16 and 15 Hz) which can be also variable as illustrated in Fig. 3.24.

This non-synchronicity of the perception functions requires the management of multi-rate information. To deal with this, we have to solve the out-of-sequence and the

temporal misalignment of the received data. Hereafter, both problems are addressed.



(a) *Perception function deviation on real-time execution. The ML lidar object detection function performs almost at a constant frequency, since its time delay process is smaller than the ML lidar sensor sampling period.*



(b) *Zoom in view of the information arrivals of the perception functions illustrating the multi-rate problem*

Figure 3.24: *Perception functions deviation and asynchronism*

3.4.1 Out-of-sequence problem

In the literature, different strategies have been proposed to ensure the right sequential data integration in a multi-sensor fusion module. These strategies differ from each other depending on the system architecture.

Let us define a standard perception function for which, three intervals are to be considered: acquisition, processing and transfer. In the acquisition step, the raw sensor data is sampled. Processing denotes the elapsed time for perception data computation. The transfer interval represents the elapsed time until the result become available by a

client application. As depicted in Fig. 3.25, a new acquisition is not triggered as long as the processing stage is not finished.

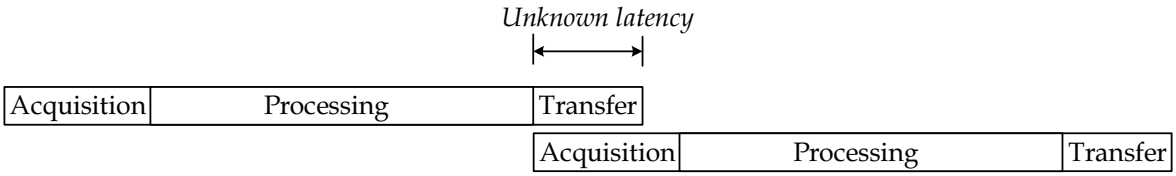


Figure 3.25: *Time intervals composing a standard perception function cycle. The transfer can occur in parallel with the data acquisition.*

In many applications, the perception functions are asynchronous and data transfers arrive at the fusion module with random delays. Each function can initiate a new acquisition in parallel to the transfer of the result report. Under these conditions, the fusion module inputs can be buffered and sorted to ensure time consistency.

In [KD03], Kaempchen and Dietmayer have examined the overall system latency for two sorting methods: a non-deterministic and a deterministic configuration. This study focused on time-critical ADAS, where the overall system latency is of high-interest.

The non-deterministic sorting method does not consider constant sensor frequencies, known latencies nor any knowledge about perception information arrivals. The principle of this technique consists in performing sorting actions only when there is, at least, one information arrival of each perception function is buffered. In this way, the information can be fused in the right order, solving the out-of-sequence arrivals.

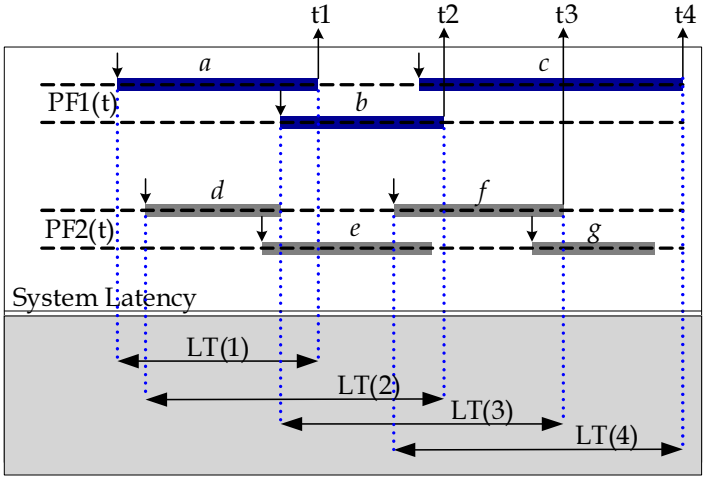


Figure 3.26: *Time arrivals example of two perception functions, $PF1$ and $PF2$, and system latency evolution, LT. Fusion sampling times are denoted $t1, ..., t4$. Incoming data instants are represented by* $\downarrow$

Consider an example of a generic multi-modal system with two data entries provided by two different perception functions denoted respectively $PF1$ and $PF2$ (see Fig. 3.26). Both functions use a common time reference. Since a new acquisition can be started

once the perception processing is finished, two parallel lines for each perception function illustrates the information sequence, as shown in Fig. 3.26.

Looking at Fig. 3.26, one can observe, at the beginning of the sequence, that the system buffers the perception reports until time $t = t1$. Then, only the oldest information is integrated, in this case $PF1(a)$. At $t = t2$, the system gets a new information from $PF1(b)$, and integrates $PF2(d)$ and $PF2(e)$. The information provided by $PF1(b)$ is integrated at $t = t3$ when $PF2(f)$ arrives. Finally, $PF2(f)$ and $PF1(c)$ are included at time $t = t4$ and $PF2(g)$ is the last integrated information of the sequence. The evolution of the system latency denoted LT is shown at the bottom of the figure.

As it could be verified in this example, the non deterministic synchronization approach satisfies the sequence order, required in the use of recursive filters like the Kalman-filter, but increases the system latency.

Alternatively, the system latency can be reduced if the transfer delays are known for at least one preceding sampling time. In this way, the system waiting constraint is relaxed, allowing the integration of buffered reports, since the arrival time of the next perception function measurement is known. This method is called *deterministic synchronization* [KD03].

Another buffer-based methodology to fuse delayed reports was implemented in the middleware architecture called AROCCAM[9]. This method proposed by Tessier et al. in [TCD+06] fuses information (in a filtering context) as it arrives providing an updated state with a minimal latency. Simultaneously, a temporal buffer stores past states and observations allowing those estimations. If an out-of-sequence observation arrives, the architecture "rewinds" the filtering process, integrates the past observation and recomputes the state integrating sorted data.

In [BP99], Blackman and Popoli studied different ways to filter out-of-sequence observations. Among the proposed approaches, a simplistic and sub-optimal solution relies on the rejection of delayed observations. This technique entails an important information loss. Another approach is to perform a "filter retrodiction" and update the filter state in the same manner as a sorted report. This approach discards process noise effects. To deal with this, more elaborate approaches are presented, but they are beyond the scope of this manuscript.

For the studied multi-modal system, the out-of-sequence problem can be solved with a deterministic synchronization technique.

[9]French acronym of *ARchitecture d'Ordonnancement de Capteurs pour la Creation d'Algorithmes Modulaires*

3.4.2 Management of the temporal data misalignment

Once the perception function arrivals are sorted, appropriate actions can be performed to integrate them in sequence. Regarding the interaction of the incoming data, one can identify two function dependencies. First, when new lidar objects are detected, ego-localization is needed to localize them in the world reference frame. Second, when a new disparity map is computed, ego-localization is also required to localize the tracked objects in the ego-vehicle frame, by compensating the movement of the mobile perception platform.

It can be remarked that, even if the report arrivals are sorted, they do not correspond to the same sampling instant. This issue constitutes the so called temporal misalignment problem. Neglecting the sampling time offsets can induce unacceptable errors in dynamic situations. To cope with this, the six ego-motion parameters, ${}^{\mathcal{S}_{t-1}}[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}]_{\mathcal{S}_t}$ have to be estimated for a prediction horizon equals to the sampling time offset.

In the following the general principle of the proposed solution is presented and then it will be applied to our special case.

General principle

The objective here is to predict a pose variation $\Delta x(t+\delta t)$ for a given temporal horizon δt. Measurements of this "ego-motion", denoted Δx_m, are available for the time interval $[t-\Delta t,\, t]$.

A first idea to get such a prediction is to consider a constant pose variation Δx proportional to the time:

$$\Delta x(t+\delta t) = \delta t \cdot \frac{\Delta x_m}{\Delta t} \tag{3.41}$$

where Δt is the sampling interval. In practice, Δx can vary a lot, so this solution seems not be good enough.

A second idea is to design a filter with an augmented state $\mathbf{x}$, which provides better Δx estimates, taking into account its possible variations. Thus, let consider the following augmented state:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \text{ with } x_2(t) = \dot{x}_1(t) \tag{3.42}$$

One can define this integrator in the discrete time state space as follows:

$$\mathbf{x}(t+1) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}(t) \text{ with } \mathbf{x}(t+1) \doteq \mathbf{x}(t+\Delta t) \tag{3.43}$$

Observations are then obtained through a linear model:

$$\Delta x_m(t) = \begin{bmatrix} \Delta t & 0 \end{bmatrix} \mathbf{x}(t) + \beta(t) \tag{3.44}$$

where the errors of this model are considered as an additive and white zero-mean Gaussian noise, denoted $\beta(t)$.

The state space system defined by Eq. 3.42, 3.43 and 3.44, is estimated through the use of a Kalman filter. In this way, one can finally obtain the prediction of the pose variation for any temporal horizon δt as follows:

$$\Delta x(t+\delta t) = \left[\begin{array}{cc} \delta t & 0 \end{array} \right] \mathbf{x}(t) \tag{3.45}$$

The above general principle can be applied straightforwardly in order to predict the ego-motion parameters. Thus, the ego vehicle state is composed by the six motion parameters and their derivatives. A linear Kalman filter with a constant accelerated model is considered, since the ego-vehicle can experience significant speed changes in braking situations. Regarding the axis-angle formalism for the attitude changes, $\Delta\boldsymbol{\omega}$, it will be remarked that for a non-holonomic moving vehicle, the variations of the attitude parameters are smooth. The latter assumption does not hold for extreme driving situations (e.g. rollover). Under these assumptions, the linear speed and the attitude changes are filtered using a single first-order model that reconstructs the derivatives of the ego-motion parameters.

Let $\mathbf{x}(t)$ be the ego state vector, $\mathbf{A}$ the state transition matrix and $\mathbf{H}$ the observation model in the following linear space state representation:

$$\begin{cases} \mathbf{x}(t) & = \mathbf{A} \cdot \mathbf{x}(t-1) + \boldsymbol{\alpha}(t) \\ {}^{\mathcal{S}(t-1)}\left[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}\right]^{T}_{\mathcal{S}(t)} & = \mathbf{H} \cdot \mathbf{x}(t) + \boldsymbol{\beta}(t) \end{cases} \tag{3.46}$$

with

$$\mathbf{x}(t) = \begin{bmatrix} \boldsymbol{\omega}(t) \\ \mathbf{v}(t) \\ \boldsymbol{\Omega}(t) \\ \mathbf{a}(t) \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} \mathbb{I}_{6\times6} & \Delta t \cdot \mathbb{I}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbb{I}_{6\times6} \end{bmatrix} \text{ and } \mathbf{H} = \left[\begin{array}{cc} \Delta t.\mathbb{I}_{6\times6} & \mathbf{0}_{6\times6} \end{array} \right] \tag{3.47}$$

where $\mathbf{v}(t)$ and $\mathbf{a}(t)$ are the linear speed and acceleration. $\boldsymbol{\omega}(t)$ represents the derivatives of the axis-angle parameters which have been considered linear to a drift $\boldsymbol{\Omega}(t)$, assumed to be constant during a sample time. In this model, the sampling interval, Δt, is not constant. The covariance of the model, denoted $\mathbf{Q}$, is chosen taking into account the errors due to the model approximation and, the covariance noise matrix $\mathbf{R}$ corresponds to a zero-mean Gaussian white noise.

By using the discrete Kalman filter equations presented in section 3.2.3, the predicted

state (*a priori*), $\mathbf{x}(t|t-1)$, and its covariance, $\mathbf{P}(t|t-1)$, are given by:

$$\mathbf{x}(t|t-1) = \mathbf{A} \cdot \mathbf{x}(t-1|t-1) \tag{3.48}$$

$$\mathbf{P}(t|t-1) = \mathbf{A} \cdot \mathbf{P}(t-1|t-1) \cdot \mathbf{A}^T + \mathbf{Q} \tag{3.49}$$

Once a new ego-motion estimation (i.e. state observation), denoted $\widetilde{\mathbf{x}}(t)$, is obtained, the predicted state and its covariance are corrected as follows:

$$\mathbf{G} = \mathbf{P}(t|t-1) \cdot \mathbf{H}^T \cdot \left(\mathbf{H}\,\mathbf{P}(t|t-1)\,\mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{3.50}$$

$$\mathbf{x}(t|t) = \mathbf{x}(t|t-1) + \mathbf{G} \cdot (\widetilde{\mathbf{x}}(t) - \mathbf{H} \cdot \mathbf{x}(t|t-1)) \tag{3.51}$$

$$\mathbf{P}(t|t) = (\mathbb{I}_{12\times 12} - \mathbf{G} \cdot \mathbf{H}) \cdot \mathbf{P}(t|t-1) \tag{3.52}$$

recalling that $\mathbf{G}$ is the optimal Kalman gain. Eq. 3.51 and 3.52 provide the *a posteriori* state and covariance.

Thus, the vision confirmation module inputs are now made of the disparity map, the ego motion parameters state and the lidar tracks. Since two of the visual confirmation module entries are filtered (ego-motion and lidar tracks), they can be predicted to correspond perfectly to the disparity map time stamp.

Using predictive filters with accurate time stamped data, the three possible processing tasks over time are listed below:

First case: A new 3D-ego localization is available. The filtered ego-motion state $\mathbf{x}(t|t-1)$ and its error uncertainty is predicted to the new observation time making use of Eq. 3.48 and Eq. 3.49. The state innovation is computed, the covariance matrix is updated and the state is corrected (Eq. 3.51 and Eq. 3.52).

Second case: A new set of objects are detected by the ML-lidar. The last known vehicle localization state is predicted up to this time. Then, the new lidar objects are localized in the world frame and assigned to the existing tracked objects. Associated tracks are updated and the remaining objects create new track hypotheses. This step called "object localization and tracking" has been detailed in previous sections.

Third case: A new disparity map is available. The vehicle localization and the tracked objects are extrapolated (i.e. predicted) at this time. Predicted objects are localized in the ego-frame using the predicted vehicle pose. In this way, the entries of the confirmation module correspond to the same instant. Finally, candidate tracks are proposed to be confirmed by making use of the disparity map.

Fig. 3.27 shows an example of possible measurements arrivals. At $t = 0$, the disparity map and the localization information are available but there is no objects to be con-

firmed. Thus, only the vehicle pose is updated. At $t = 1$, a set of objects have been detected. They are localized using the predicted vehicle pose. At $t = 2$, tracks can be confirmed using the predictions of the objects and the vehicle pose.

We remarked during our experiments that this mechanism dramatically increases the performance of the system, particularly when the scene includes a significant dynamic content.

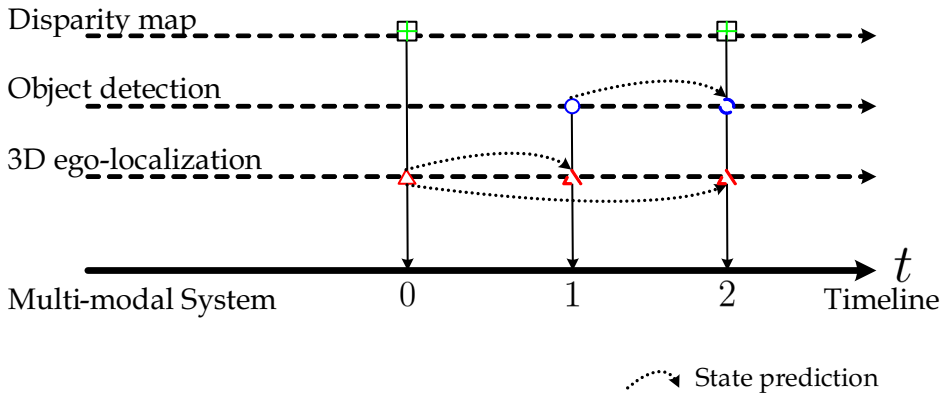


Figure 3.27: *Example of possible data arrival*

3.5 Experiments

An objective of the trial reported in this section is to provide a proof concept of the proposed track confirmation methodology, using real data.

It is worth mentioning that real data tests in a multi-sensor platform requires a great effort to be accomplished, since other tasks must be ensured beforehand, for instance, the calibration procedures of the SVS cameras and the SVS/ML-lidar and the implementation of the middleware modules for the real-time data acquisition system.

It should be recalled that the visual track confirmation relies not only on the image-depth information (i.e. ROI disparity) but also, on data coming from asynchronous fusion modules (i.e. ego-motion and object tracking).

The perception modules were implemented for real-time execution and provide essential data for the computation of the 3D dynamic perception map. With the aim of simplifying the experimental process, the out-of-sequence problem was solved by post-processing. Thus, the perception modules outputs were precisely logged and their arrivals were sorted in time. Even though the implementation is examined in offline conditions, it still remains the temporal misalignment of the reports which is dealt with through the use of predictive filtering. The proposed object tracking and visual confirmation algorithms were implemented in Matlab as depicted in Fig. 3.28.

The experiment, reported here, was performed using the real data set called as “closed-loop data set” in section 2.9 of chapter 2, which was also used for the evaluation of the

multi-modal ego-motion estimation method. This data set consists in three different trials following the same trajectory. Two of these trials provide some common traffic situations where the ego-vehicle and other objects (e.g. pedestrians and moving and parked vehicles) interact together in a urban scene. In the third test sequence, almost none moving objects were observed.

In the remaining of this section the obtained results will be assessed starting with the object tracking function. At the last, the visual track confirmation results are analyzed.

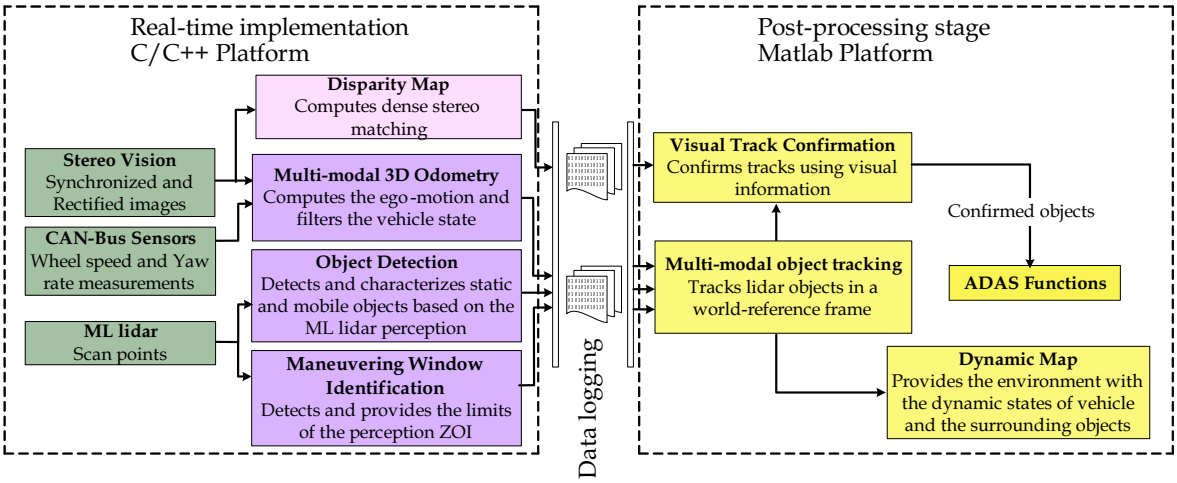


Figure 3.28: *System implementation for the experimental trial of object tracking and visual track confirmation. The multi-modal sensing functions: Object detection, maneuvering window identification, disparity maps computation and the 3D multi-sensor ego-localization were processed in real time and their outputs were logged. The multi-modal object tracking and the track confirmation function were implemented in Matlab. The results were obtained in offline conditions, taking into account the temporal misalignment of the logged reports.*

3.5.1 Maneuvering Window Identification

An interesting feature of the multi-modal system is its capability to characterize a maneuvering window where potential obstacles require a special attention. Thus, the evolution of this estimated region was reconstructed over the complete trajectory. Fig. 3.29 illustrates the bird-eye view of the reconstructed zone of interest (i.e. ${}^{\mathcal{W}}XZ$ plane) in the fixed-reference frame.

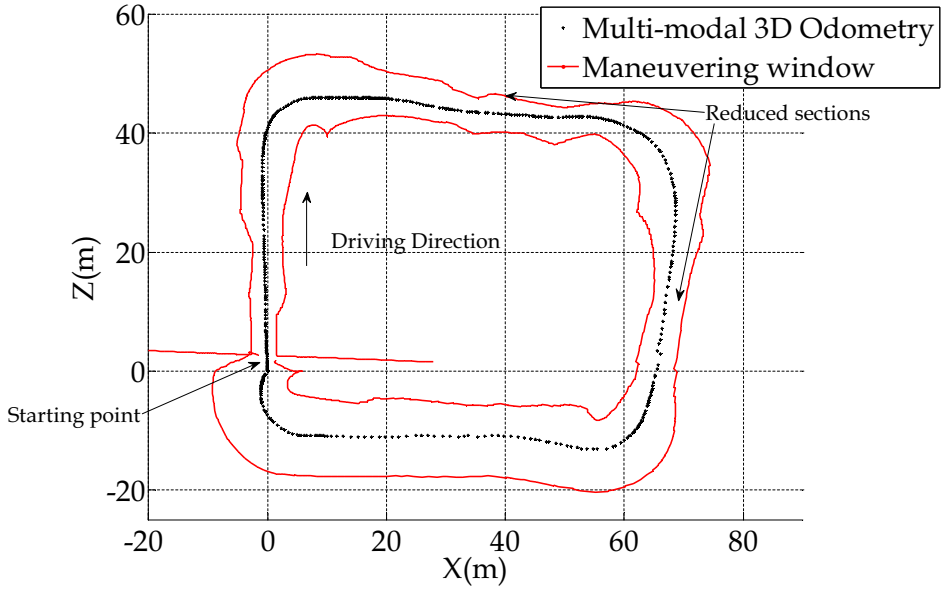


Figure 3.29: *Reconstruction of the maneuvering window* $\left({}^{\mathcal{W}}XZ \text{ plane view}\right)$. *It can be noticed the convergence of the filter after initialization. Reduced sections are due to parked vehicles.*

This maneuvering window reconstruction was obtained by exchanging geometrical data between the lidar and the vision-based functions thanks to the calibration information. It can be remarked that, at the beginning of the test sequence (i.e. starting point (0,0)), the vehicle remains stationary, which shows how the filtered boundaries of the maneuvering window converges. Along the trajectory, some parked vehicles were observed. Those trajectory sections were highlighted by the reduction of the identified zone. These results constitute a very interesting feature which can certainly be coupled to a GIS (Geographic Information System) for map-matching and global localization applications.

3.5.2 Object Localization and Tracking Results

Focusing now on the kinematic state estimation of the tracked objects, Fig. 3.30 illustrates a zoomed area of the dynamic map. The area herein shows some state samples of a tracked vehicle and its corresponding track re-projection on the left SVS camera image. At the bottom of the figure, the size of the track is represented by its bounding circle, in red, and its center as a red triangle. The corresponding image track projections and their speed vectors are also illustrated in the upper part of the figure. The projected bounding box encloses the 3D cylinder of the track as shown in Fig. 3.18.

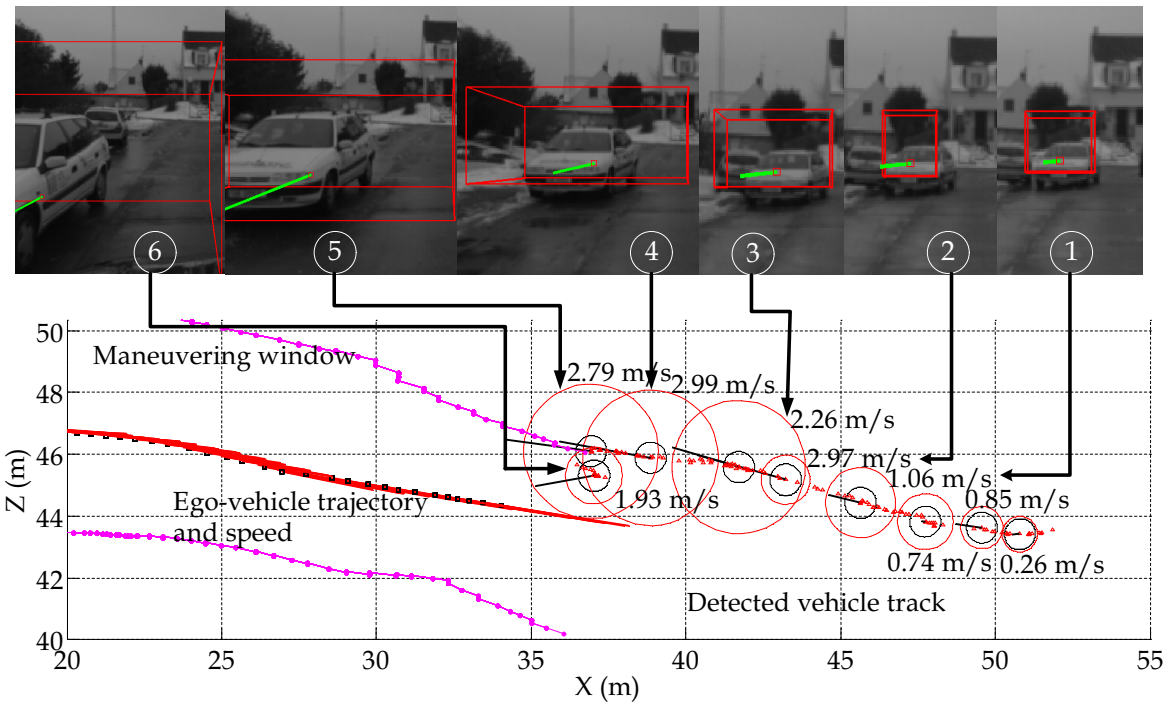


Figure 3.30: *Trajectory of a tracked vehicle. In the image sequence: Track projections are represented by 3D red boxes and the track speed vector by green lines. In the dynamic map zoomed area: The location is illustrated in six sample times. Red triangles denote the estimated centroids of the tracked object over time. The detected track size is illustrated by red circles. Black circles represents track covariance uncertainty. The ego-vehicle state evolution is also provided.*

Additionally, Fig. 3.30 also shows how the detected track size changes as the object surface is impacted by the ML lidar. This fact sometimes entails perturbations in the speed estimation, since important size changes induce a spurious motion of the track centroid. This problem has been already addressed in [PT09, FC07], where the spurious centroid displacement and the occlusions were dealt using a model based vehicle tracking. Looking at the image projection of the track speed vector, however, one can see that, the multi-modal system performs quite well. It is worth mentioning that objects are tracked even if they come out of the SVS field-of-view.

Fig. 3.31 shows another section of the dynamic map. Indeed, estimating the speed of a pedestrian is a challenging task, since its motion is, sometimes, unpredictable. However, a linear motion at constant speed has shown to be here very pertinent.

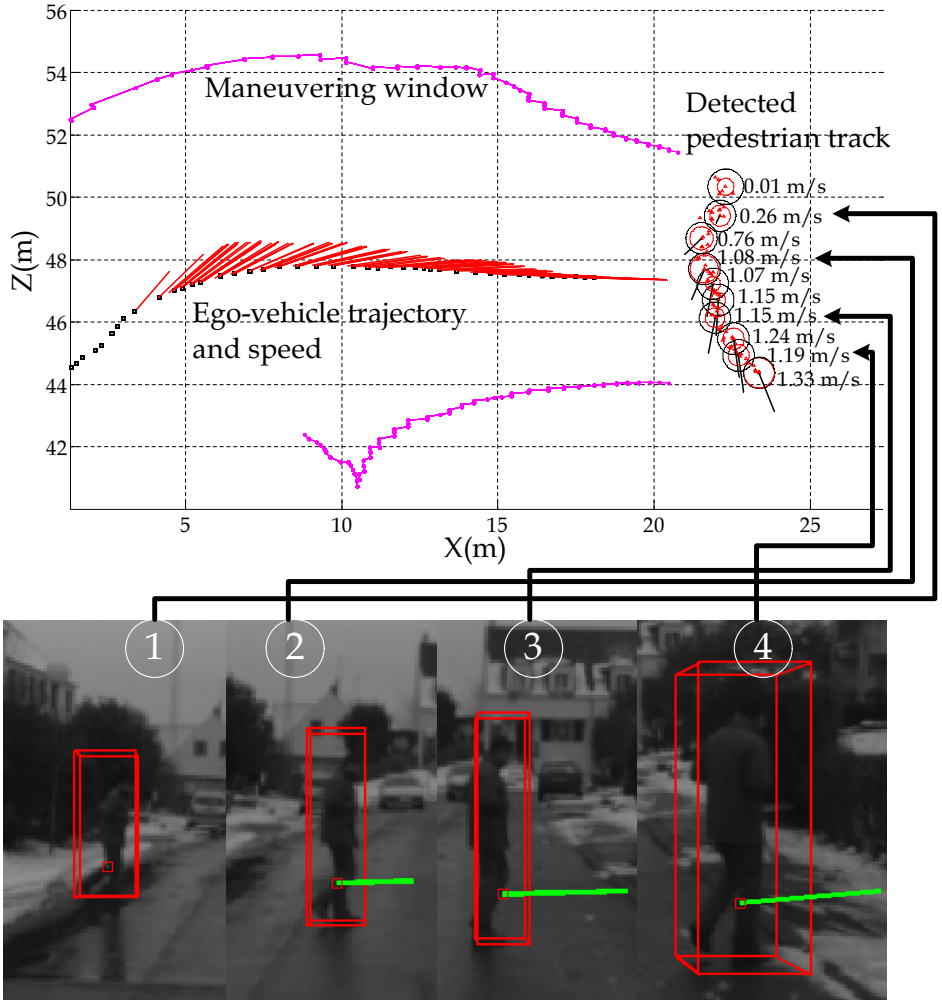


Figure 3.31: *Trajectory of a pedestrian when crossing a road. The figure illustrates the trajectory, the speed and the detected size of a pedestrian. The conceived system provides smooth speed estimates of the tracked object.*

A usual question here would be, how to establish an object trajectory ground truth? For this, a distributed data logging system with a common timeline reference is required. As this system being not available in the laboratory, no ground truth for the track localization was used during the experiment. However, the reconstructed trajectory illustrated in Fig. 3.31 is still meaningful, since it corresponds quite well to the pedestrian's observed trajectory in the snapshot sequence. Moreover, the estimated speed magnitudes are also coherent to walking speed average accorded to a pedestrian [SW09].

In Fig. 3.32 a wheelchair pedestrian is successfully tracked. The tracking system is able to estimate very small motion variations thanks to the accurate measurements of the ML lidar and the good estimation of the vehicle displacement.

The fast convergence of the track filtering grants access to a good kinematic object estimate even when the vehicle has a significant rotation speed, as shown in Fig. 3.33. However, a degradation of the speed vector direction is noticeable when objects are

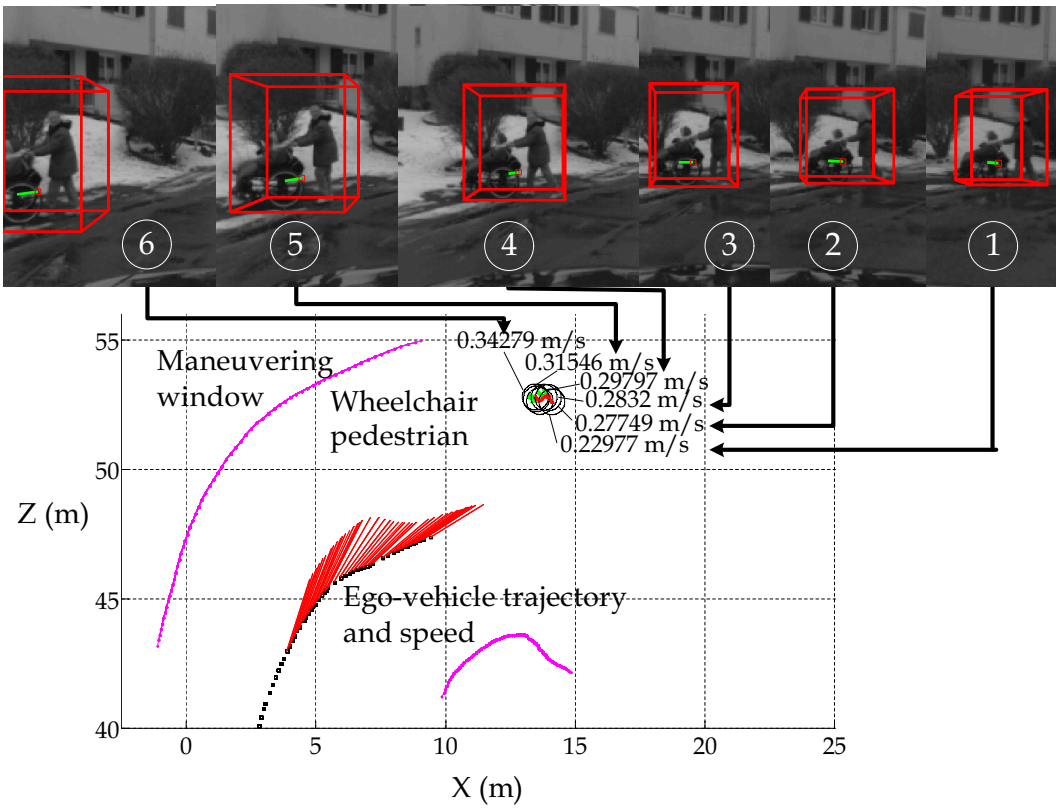


Figure 3.32: *Wheelchair pedestrian trajectory observed from the moving vehicle in a turn*

getting out of the field-of-view of the sensor (see the two last samples).

3.5.3 Visual Confirmation of Mobile Objects

Performance indicators for the analysis of the considered system

As stated before, the visual track confirmation constitutes a self-assessment functionality aimed at increasing the integrity of the tracks outputted by the object detection system.

The object detection module can deliver True Positive (TP) and True Negative (TN) outputs which constitute indicators of a correct detection. Nevertheless, it can also deliver False Negative (FN) outputs representing missed object detections and False Positives (FP) which stand here for false alarms. This module being a detector and not a classifier, the quantification of TN outputs is not possible (it would require determining how many objects were not detected when no objects were observed). Since visual confirmation can only test the existence of detected objects, the FN indicator of the object detection is not relevant here. In consequence, only detected objects manually classified as TP or FP are pertinent in the evaluation of the visual track confirmation.

Looking at the visual confirmation output, one can identify as well TP, FP, TN and FN. True Positives represent detected objects being successfully confirmed, which quanti-

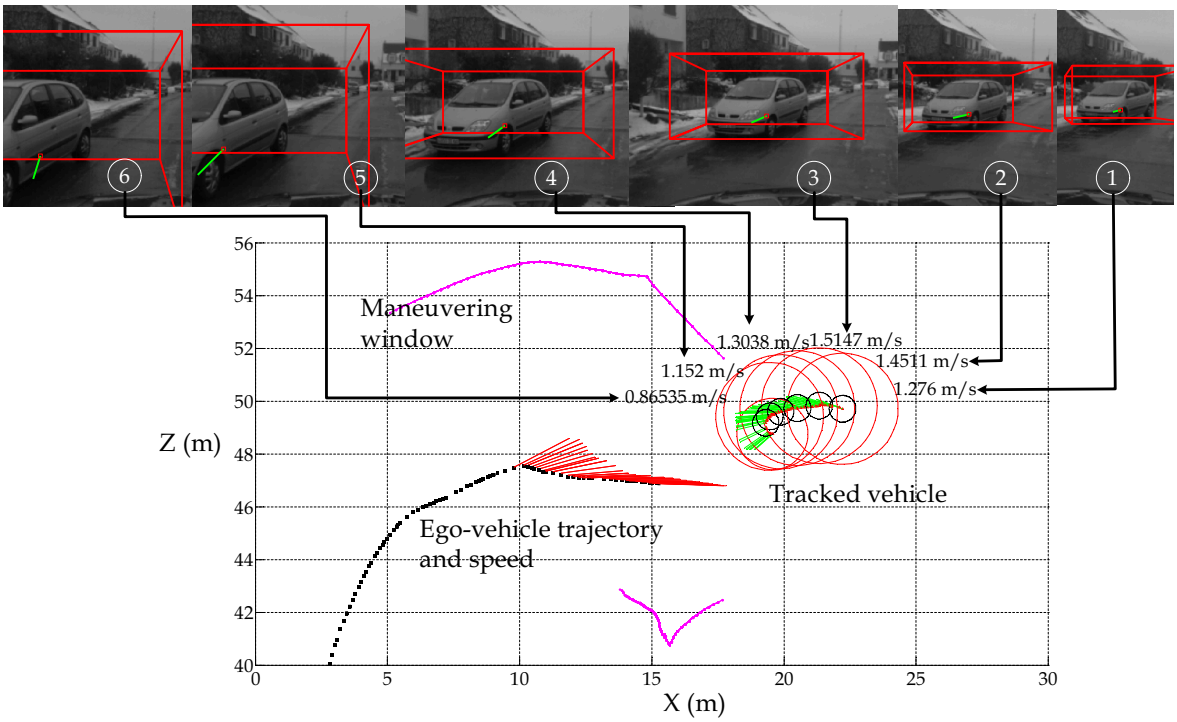


Figure 3.33: *Trajectory of an observed vehicle while the ego-vehicle is taking a turn.*

fies the confirmation rate. On the contrary, False Negatives stand for detected objects which have not been confirmed, reducing the availability of the detection function. False positives of the visual confirmation refer to spurious objects which are erroneously confirmed and True Negatives quantifies the rejection rate of the detection False Alarms.

As TP and FP are complementary indicators to FN and TN respectively, the performance of the visual track confirmation is assessed through the quantification of the false alarms rejection and the track confirmation rate. In the following the evaluation methodology is detailed and the obtained results are reported.

False Alarm Rejection Rate Quantification

The performance of a binary classifier is generally evaluated by the means of a ROC[10] curve [Faw06]. To this end, ground truth is required to compute TP and FP indicators. During the experimental test, it was, however, not possible to extract enough false detections from real data reports to compose a representative ground truth sample. This is because false alarms in real situations constitute sparse random artifacts (e.g. segmentation errors, spurious lidar measurements like rain and snowflakes).

In consequence, the false alarm rejection rate was evaluated by providing manually a phantom track placed four meters in front of the vehicle at every sample time (which means that the lidar is providing a 100% FA rate). Using a trial sequence where no

[10] *Receiver Operating Characteristic*

objects on the road were observed, the visual confirmation function only dealt with these spurious tracks. Five false alarm conditions have been computed by changing the phantom track size from 80cm to 2m. The results are shown in Table 3.2.

The computation of ROC curve from this FP indicator is not appropriate, since it does not represent the global performance gain of the system. However, it remains meaningful regarding to the single performance of the visual confirmation module. Moreover, one can highlight that the reported false alarm rejection rates were obtained in the worst-case conditions (i.e. 100% of false alarm input) which is unlikely in practice.

The track size changes seem to do not have an important influence in the performance of the visual track confirmation as shown in Table 3.2.

Table 3.2: *False Alarm Rejection Rate Quantification*

Spurious Track size (m)	0.8	1	1.2	1.5	2
Confirmed spurious measurements	170	207	208	219	184
Number of Analyzed Frames	2856	2856	2856	2856	2856
False Alarm Rejection Rate	94.04	92.75	92.72	92.33	93.55
Erroneous Confirmations (%)	5.95	7.24	7.28	7.66	6.44

This test has also revealed that the principal source of incorrect confirmations are the erroneous disparity map regions of the ROI. An example is shown in Fig. 3.34.

Figure 3.34: *Phantom track erroneously confirmed*

As the proposed method can filter up to 93% of false alarms, these results are quite encouraging.

Visual Track Confirmation Rate Quantification

Another main objective of confirming tracks is to preserve the true positive detection rate, ensuring in this way the availability of the object detection function. In order for

the confirmation performance of the system to be evaluated, we report experimental results of the visual confirmation function using five sequences.

Use Cases and Evaluation Methodology These use cases are relevant to common scenarios in urban environments. Fig. 3.35 gives a graphical description of the considered situations involving three kinds of mobile objects: pedestrians, wheelchair pedestrians and cars.

In the reported experiments, the ML-lidar did not give rise to any missed detections. The evaluation methodology aims at quantifying the percentage of time during which the object tracking function becomes unavailable because of visual non-confirmations.

The ground truth was referenced manually in the left image plane of the SVS. In this way, the center point coordinates of the observed objects of interest were selected, frame by frame. All the objects considered in the ground truth were localized in a common perception region for the SVS and the ML-lidar. The confirmation track rate was computed by counting the number of times where the bounding box of the confirmed track contains the ground truth point.

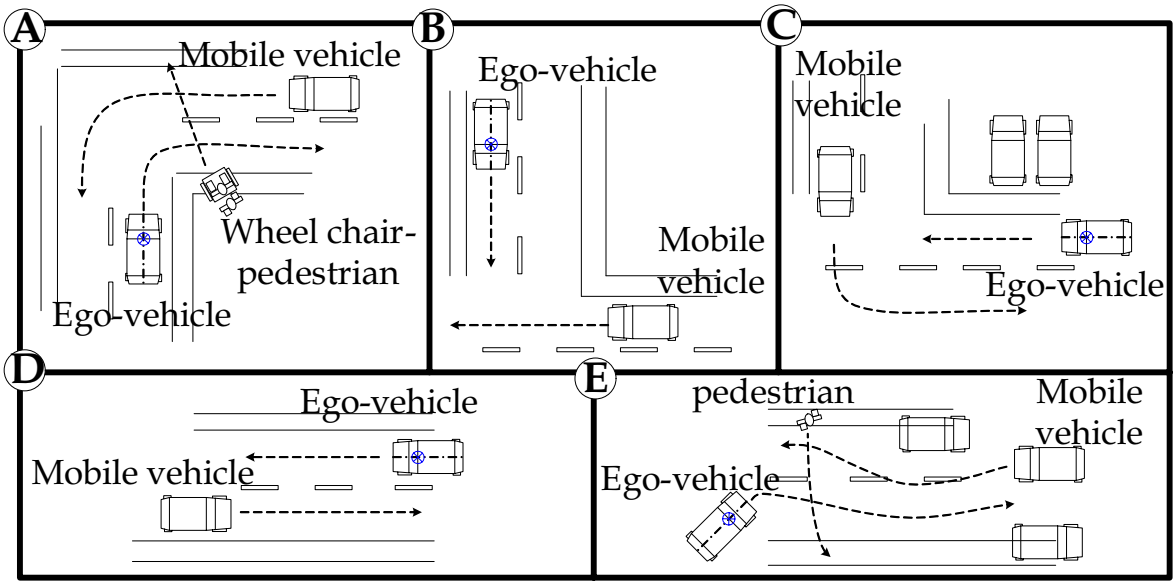


Figure 3.35: *Use cases considered in the evaluation test*

The results obtained using the ground truth are reported in Table 3.3. A total of 650 frames in 5 different situations showed that, at least 81% of the time, the detected objects of interest were confirmed by the two modalities. Although one may conclude that the visual track confirmation may sometimes reduce the number of true positives, it should nevertheless be remarked that the confirmed tracks ensure the integrity of the complete perception process.

Additionally, it should be remarked that visual track confirmation has no influence in the iterative object tracking cycle which means that even if a track is not confirmed,

the system keeps tracking it.

Table 3.3: *Rate of detected objects confirmed by stereo vision*

Video Sequence	A	B	C	D	E
Duration (s)	5	4	5	6	10
Number of Analyzed Frames	110	90	90	125	235
Number of scans	78	51	62	94	137
Positioning updates	72	37	65	121	160
Visual Confirmation Rate (%)	100	100	81.8	98.5	83.5

In use cases C and E, it was observed that large changes in vehicle pitch angle can influence the precision of object tracking, since object motion is considered to be planar and the vehicle pitch angle w.r.t. the ground is unknown. These effects are shown in Fig. 3.36 where it can be observed that the ROI has a slight delocalization in height.

Figure 3.36: *Confirmed vehicle in scenario C using stereo vision. The figure shows a vehicle having just turned. Here the ROI is delocalized because of pitch changes, given that the ego-vehicle is accelerating (see Fig. 3.35). In the right-hand side, the disparity map of the analyzed scene is provided.*

Fig. 3.37 illustrates the effectiveness of the confirmation concept and the accuracy in the estimation of the speed track direction.

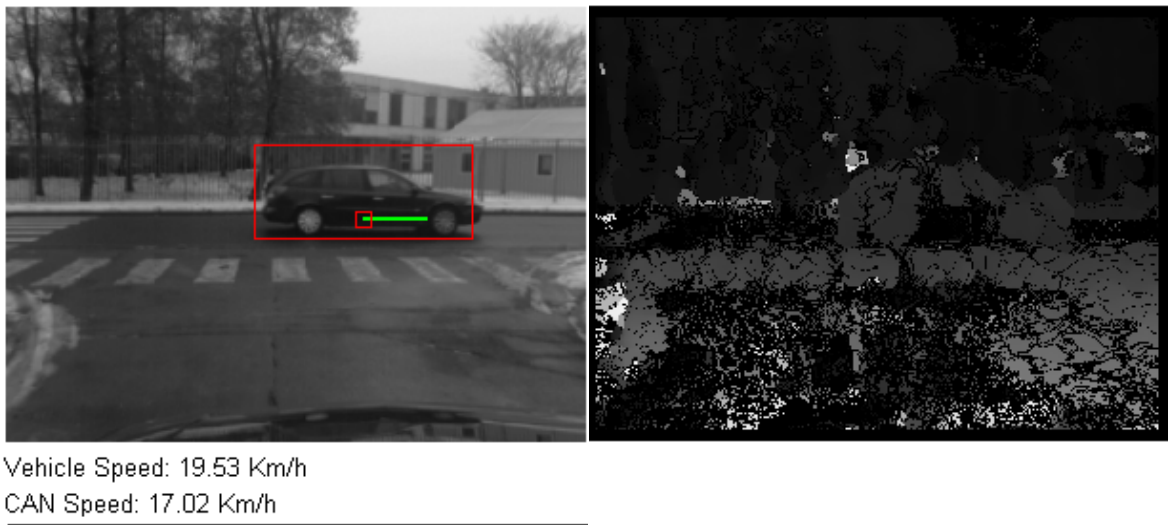


Figure 3.37: *Vehicle visually confirmed in the intersection included in the scenario B (see in Fig. 3.35). The ROI shows a quite good localization which would be interesting for recognition tasks.*

It should be reminded that absolute track speed direction can be estimated thanks to the use of an object tracking approach based on a fixed-reference frame.

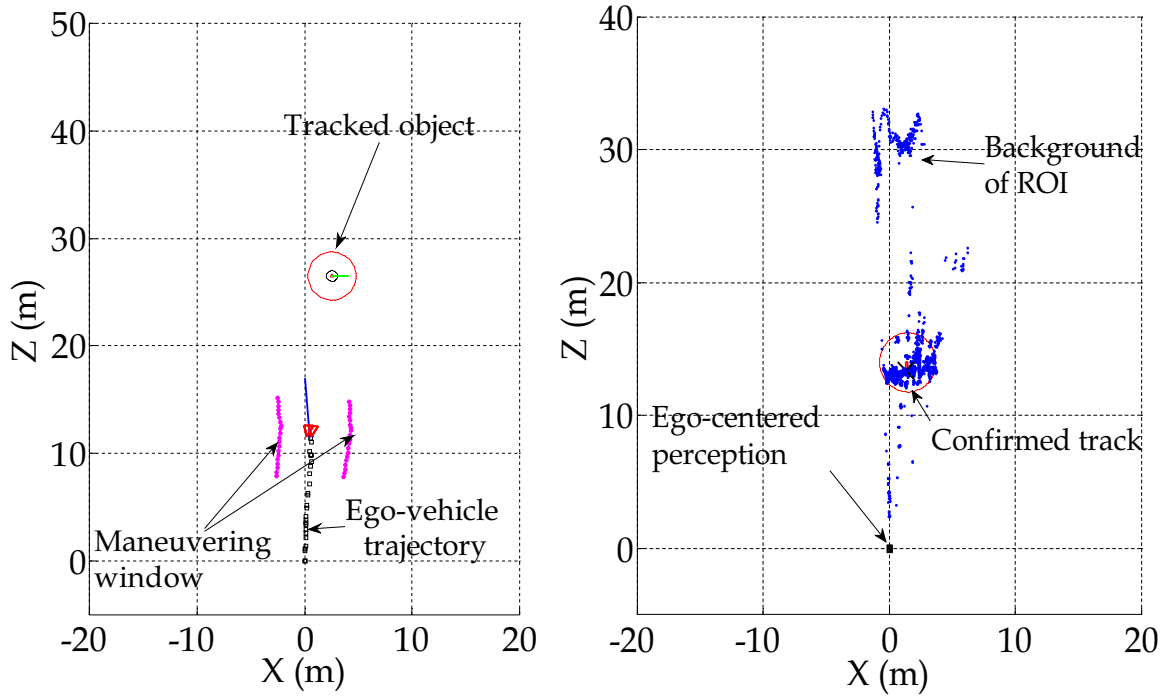


Figure 3.38: *Example of a confirmed tracked object using the SVS. In the left-hand side is shown the world map containing a tracked object within the maneuvering window. In the right-hand side, the tracked object is confirmed, since one of the cluster centroid matches the lidar estimated position.*

In Fig. 3.38, the left-hand side illustrates the world map, corresponding to Fig. 3.37, where the ego-vehicle and the detected objects are localized and tracked. The right-hand side of the figure shows, in blue, the reconstructed points of the ROI image.

Looking at the ego-map, it could be remarked that one of the centroids of the clustered point cloud has been matched with the track. This match confirms the existence of the detected object.

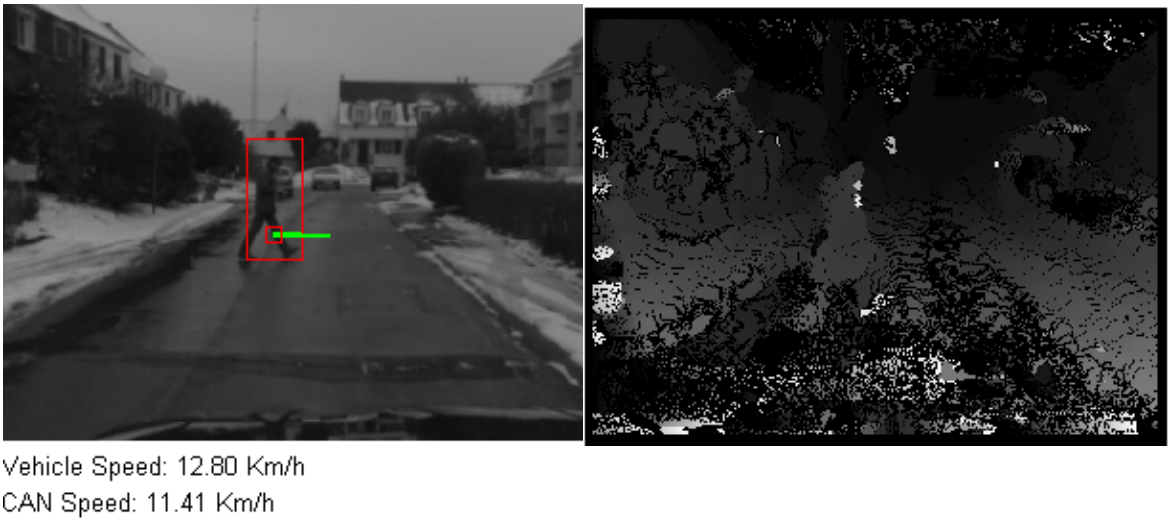


Figure 3.39: *A confirmed pedestrian track. The figure shows the confirmed pedestrian when crossing the road in Scenario E (Fig. 3.35)*

Fig. 3.39 presents another example of a confirmed object. Its bounding box (in red) and its speed vector projection (in green) show a good localization. Thus far, these results validate the synchronization strategy in scenarios involving slow and fast moving objects. Finally, the 3D view of the dynamic map corresponding to the instant illustrated in Fig. 3.39, is provided in Fig. 3.40.

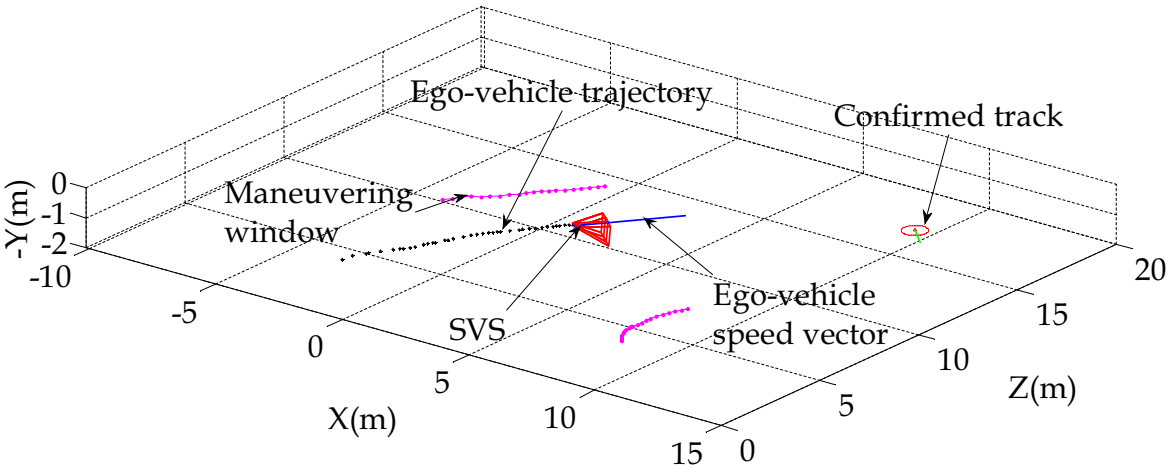


Figure 3.40: *Dynamic map of a pedestrian tracked by the multi-modal system.*

3.6 Conclusion

An asynchronous embedded multi-modal system for object localization and tracking has been proposed and experimentally tested. The approach presented here provides a

3D *dynamic* map of a vehicle's surroundings. The method merges sensor data to achieve a robust, accurate 3D ego-localization. This function is combined with a lidar-based object tracking focused on a maneuvering window, which provides the trajectories and speeds of objects as they move in the space. A visual confirmation function integrated into the tracking system checks the integrity of the provided information. To this end, ROI are processed in a dense way. 3D points are reconstructed and compared to the lidar-tracked object. This scheme makes full use of the broad functional spectrum of stereo vision systems. Synchronization issues are taken into account to ensure the temporal consistency of the system.

The obtained results show the effectiveness of the proposed strategy to increase integrity. The visual confirmation function was tested experimentally, demonstrating a good confirmation rate and a good false alarm rejection. Additional tests are necessary to provide more complete information about the system performance under different scenarios.

The inclusion of a visual object recognition function for selecting the most suitable object motion model might improve the tracking process. This is a research outlook.

Conclusions and Outlook

Conclusions

This research was devoted to the study of some contributions of stereo vision systems for multi-modal perception of the geometry of the environment. We mainly focused on multi-sensor object localization and tracking for Intelligent Vehicles evolving in open roads.

Accordingly, this manuscript addressed multi-sensor calibration, visual ego-motion estimation in dynamic environments, multiple target tracking for ADAS applications and perception integrity using vision.

The conducted investigations have led us to establish conclusions based on experimental results. Hereafter, we present what we believe to be our contributions.

In order to address the cooperative fusion of the geometrical scene perception between a ML lidar and a vision system, their relative pose has to be estimated accurately. To this end, a new extrinsic calibration method taking into account some specificities of vehicle integration has been proposed. This method relies on a robust registration of multiple pose observations performed on a dedicated target. Observations are simultaneously obtained by the ML lidar and the camera. The circle-based calibration target allows the estimation of not only the extrinsic system parameters but also the intrinsic camera ones. Conscious of the importance of quantifying errors in data sensor fusion applications, we also computed the confidence intervals of the extrinsic parameters. We deeply studied the stability and the performance of the proposed solution in simulated and real conditions. The geometrical transformation that is obtained allows an accurate projection of the lidar data onto images. Moreover, we have noticed that errors in the intrinsic camera parameters are compensated by the extrinsic ones, reducing the offsets between measurements in the 3D space.

Concerning the dynamic object motion analysis of the scene, we developed a world-centered approach because it simplifies the tracking task by improving the kinematic state estimation of the observed objects. Nevertheless, this approach requires precise knowledge of the ego-motion. To this end, we proposed a real-time visual odometry method which was considerably improved by the use of wheel speed sensors and gyrometer measurements. This multi-sensor strategy copes with vision system limitations in

high rotational speed motions like 90° turns and roundabouts. Additionally, a good trade-off between precision and execution time was achieved thanks to a sparse feature approach and rigidity scene constraints provided by a quadrifocal tensor warping.

Making use of the full 3D vehicle pose estimate and a simultaneous lidar-based object detection, a dynamic map of the vehicle's surroundings can be created. Experimental results confirmed how likely collisions can be straightforwardly detected based on trajectories and speed vectors of the observed objects. Since the multiple target tracking is a complex problem particularly in urban environments, the approach deals with tracks only within a maneuvering window identified by the means of the lidar.

Keeping in mind that the dynamic map is intended to provide relevant information to ADAS, we oriented our efforts towards the development of a self-assessment function able to check the integrity of the track hypothesis. Thus, a visual confirmation function was integrated to check the real existence of the tracked objects. Our tests carried out on a full scale automotive system showed a good performance which preserves the high detection efficiency of the system. Additionally, synchronization issues of the multiple perception modalities were also considered and a solution was deployed through the use of predictive filters which ensures the temporal data fusion consistency.

Future research

Along this document, we explored different areas related to a challenging objective. We obtained some promising results in experimenting our suggested solutions. We identified some problems and interesting clues which worth pursuing in a future research.

A first perspective is the quantification of measurements errors and their propagation in the fusion system in order to determine confidence indicators of the provided information. Such indicators evidence the trust accorded by the driver to the fused data. In this thesis, this topic was only examined for the extrinsic calibration parameters. Preliminary tests (not reported in this manuscript) performed for quantifying the confidence of the visual odometry estimates were shortly studied through the use of the Unscented Transform (UT) [Jul02] to propagate residual errors to the pose state. The UT transform seems to be a good candidate since it allows computing efficiently the prediction of means and covariances in nonlinear systems. Further study is necessary particularly when the visual optimization fails.

A second research perspective concerns global localization of visual odometry. In chapter 2, a complete routine has been presented to obtain a GPS-like positioning using visual odometry, in situations where GPS is not reliable. This technique could be enhanced by integrating a Geographical Information System (GIS) with a map matching algorithm using vectorial or aerial image information. In this way, the localization method based on visual odometry can be initialized even in inaccurate GPS conditions.

Moreover, drifts typically observed in dead-reckoning techniques could be also detected and corrected. In order to do such a strategy, the use of a confidence information associated with the visual odometry is of prime importance.

Finally, we can mention some improvements regarding the object tracking scheme presented in chapter 3. One of them is the inclusion of a visual object recognition function coupled with an interacting multiple model filtering. This would increase the accuracy of the Kalman filter predictions and would facilitate the assignment of better geometrical object models, like for instance, cylinders for pedestrians and boxes for vehicles. This latter betterment should deeply improve the object track assignment and the lidar-vision data fusion. The development of an object detection solely based on vision is also a perspective of this work, since it would provide a redundant detection system to the fusion architecture.

Appendix A

Rigid-body Transformations

A.1 Introduction

This study makes use of the concepts of the three-dimensional Euclidean space, $\mathbb{E}^3$, defined by the axioms of Euclid. This space is represented by a Cartesian coordinate frame where every point $p \in \mathbb{E}^3$ is identified by a point in $\mathbb{R}^3$ with three coordinates:

$$\mathbf{p} \doteq \mathbf{p}_{(x,y,z)} = \begin{bmatrix} \mathbf{p}_x & \mathbf{p}_y & \mathbf{p}_z \end{bmatrix}^T = \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \\ \mathbf{p}_z \end{bmatrix} \in \mathbb{R}^3 \tag{A.1}$$

Sometimes, we can also use two subscripts to indicate 2D coordinates such as $\mathbf{p}_{(x,y)} = \begin{bmatrix} \mathbf{p}_x & \mathbf{p}_y \end{bmatrix}^T$ for example. Since we established a one-to-one mapping between $\mathbb{E}^3$ and $\mathbb{R}^3$, we can say that points and their coordinates are equivalent concepts.

Multiple sub-frames can be defined within the Cartesian space. Frames are then denoted by calligraphic letters (e.g. $\mathcal{W}$, $\mathcal{S}$, $\mathcal{C}$). A point defined in a specific frame is referenced with a left superscript indicating that frame e.g. ${}^{\mathcal{W}}\mathbf{p}$.

Consider now a point ${}^{\mathcal{C}}\mathbf{p}$ referenced in a frame $\mathcal{C}$ which position and attitude are defined in the world frame $\mathcal{W}$ by ${}^{\mathcal{W}}[\mathbf{R}, \mathbf{t}]_{\mathcal{C}}$ as illustrated in Fig. A.1. ${}^{\mathcal{W}}[\mathbf{R}, \mathbf{t}]_{\mathcal{C}}$ can be used as a rigid-body transformation. So, one can highlight that left superscript denotes the "destination frame" and the right subscript indicates the "origin frame". The notation ${}^{(\cdot)}[\cdot, \cdot]_{(\cdot)}$ is not a matrix representation but an abstract notation as we will see later.

From above and in order to define the coordinates of ${}^{\mathcal{C}}\mathbf{p}$ in the world frame, the rigid-body transformation which has to be applied to it is given by,

$${}^{\mathcal{W}}\mathbf{p} = {}^{\mathcal{W}}\mathbf{R}_{\mathcal{C}} \cdot {}^{\mathcal{C}}\mathbf{p} + {}^{\mathcal{W}}\mathbf{t}_{\mathcal{C}} \tag{A.2}$$

where the variable $\mathbf{R}$ is reserved to denote rotation matrices belonging to the special

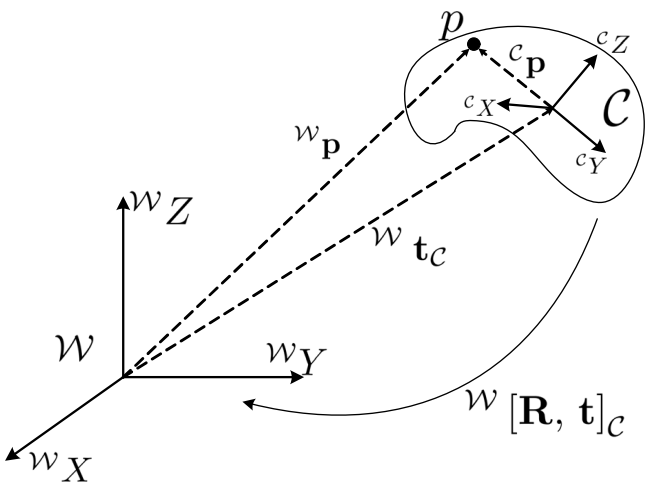


Figure A.1: *Rigid-body transformation between a frame $\mathcal{C}$ and a world frame $\mathcal{W}$*

group $SO(3)$ of orthogonal matrices in $\mathbb{R}^{3\times3}$ and satisfying:

$$\begin{aligned} \mathbf{R}^T\mathbf{R} &= \mathbb{I}_{3\times3} \\ \det(\mathbf{R}) &= +1 \end{aligned}$$

Translation vectors are denoted by $\mathbf{t}$ in $\mathbb{R}^3$.

Additionally, Eq. A.2 implies that a point $\mathbf{p}$ referenced in the world frame $\mathcal{W}$ can be transformed to the frame $\mathcal{C}$ applying an inverse transformation:

$$\begin{aligned} {}^{\mathcal{C}}\mathbf{p} &= {}^{\mathcal{W}}\mathbf{R}_{\mathcal{C}}^T\left({}^{\mathcal{W}}\mathbf{p} - {}^{\mathcal{W}}\mathbf{t}_{\mathcal{C}}\right) \\ &= {}^{\mathcal{W}}\mathbf{R}_{\mathcal{C}}^T\,{}^{\mathcal{W}}\mathbf{p} - {}^{\mathcal{W}}\mathbf{R}_{\mathcal{C}}^T\,{}^{\mathcal{W}}\mathbf{t}_{\mathcal{C}} \end{aligned} \tag{A.3}$$

where the rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{C}}^T$ is equivalent to ${}^{\mathcal{C}}\mathbf{R}_{\mathcal{W}}$, and the operand $\left(-{}^{\mathcal{W}}\mathbf{R}_{\mathcal{C}}^T\,{}^{\mathcal{W}}\mathbf{t}_{\mathcal{C}}\right)$ represents the translation vector ${}^{\mathcal{C}}\mathbf{t}_{\mathcal{W}}$.

A.2 Rotation representations

Hereinbefore, rotations were presented and considered in their 3×3 matrix representation. However, there are multiple representations for the rotations which can be used such as: Euler vectors, quaternions and axis-angle vectors.

A.2.1 Euler Angles

A rotation can be represented by Euler angles $\mathbf{e} = \begin{bmatrix} \phi & \beta & \psi \end{bmatrix}^T$ which correspond respectively to roll, pitch and yaw orientations in the ZYX convention. Since an Euler angle vector cannot be used to perform direct transformations, it has to be converted into a classical rotation matrix representation. Hence, the rotation matrix for the given

Euler angles (in this order) is:

$$\mathbf{R} = \begin{bmatrix} \cos\beta\,\cos\psi & \sin\phi\,\sin\beta\,\cos\psi - \cos\phi\,\cos\psi & \cos\phi\,\sin\beta\,\cos\psi + \sin\phi\,\sin\psi \\ \cos\beta\,\sin\psi & \sin\phi\,\sin\beta\,\sin\psi + \cos\phi\,\cos\psi & \cos\phi\,\sin\beta\,\sin\psi - \sin\phi\,\cos\psi \\ -\sin\beta & \sin\phi\,\cos\beta & \cos\phi\,\cos\beta \end{bmatrix} \tag{A.4}$$

A.2.2 Quaternions

An alternative to represent rotations is to use unit quaternions. Hereafter, only a summary is provided.

A unit quaternion is a denoted by

$$\mathbf{q} = q_0 + q_1 i + q_2 j + q_3 ij = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \tag{A.5}$$

where q_0, q_1, q_2 and q_3 are real numbers and the 4-vector $\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$ is unit length. The conjugate of $\mathbf{q}$ is defined as follows:

$$\bar{\mathbf{q}} = q_0 - q_1 i - q_2 j - q_3 ij = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}^T \tag{A.6}$$

The corresponding rotation matrix of $\mathbf{q}$ is given by:

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\cdot(q_2 q_3 - q_0 q_1) & 2\cdot(q_1 q_3 + q_0 q_2) \\ 2\cdot(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2\cdot(q_2 q_3 - q_0 q_1) \\ 2\cdot(q_1 q_3 - q_0 q_2) & 2\cdot(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{A.7}$$

In opposition to Euler angles and axis-angle representations, quaternions can be used to perform direct transformations. Let ${}^{\mathcal{W}}[\mathbf{q},\,\mathbf{t}]_{\mathcal{C}}$ be the transformation that corresponds to ${}^{\mathcal{W}}[\mathbf{R},\,\mathbf{t}]_{\mathcal{C}}$ but which uses quaternions to express rotations. Accordingly, Eq. A.2 can be computed as follows:

$${}^{\mathcal{W}}\underline{\mathbf{p}} = {}^{\mathcal{W}}\mathbf{q}_{\mathcal{C}} \star^{\mathcal{C}} \underline{\mathbf{p}} \star^{\mathcal{W}} \bar{\mathbf{q}}_{\mathcal{C}} + {}^{\mathcal{W}}\underline{\mathbf{t}}_{\mathcal{C}} \tag{A.8}$$

where $\star$ represents the multiplication quaternion operator. Underlined vectors (e.g. $\underline{\mathbf{p}}$) denote expanded forms (i.e. $\underline{\mathbf{q}} = [0,\,\mathbf{p}]^T$) for the use of the quaternion multiplication.

Quaternion multiplication

Given two unit quaternions $\mathbf{q1} = \begin{bmatrix} q1_0 & q1_1 & q1_2 & q1_3 \end{bmatrix}^T$ and $\mathbf{q2} = \begin{bmatrix} q2_0 & q2_1 & q2_2 & q2_3 \end{bmatrix}^T$, the product is defined by

$$\mathbf{q1} \star \mathbf{q2} = \begin{bmatrix} q1_0\, q2_0 - q1_1\, q2_1 - q1_2\, q2_2 - q1_3\, q2_3 \\ q1_0\, q2_1 + q1_1\, q2_0 + q1_2\, q2_3 - q1_3\, q2_2 \\ q1_0\, q2_2 - q1_1\, q2_3 + q1_2\, q2_0 + q1_3\, q2_1 \\ q1_0\, q2_3 + q1_1\, q2_2 - q1_2\, q2_1 + q1_3\, q2_0 \end{bmatrix} \tag{A.9}$$

It should be noticed that $\mathbf{q} \star \bar{\mathbf{q}} = \bar{\mathbf{q}} \star \mathbf{q} = 1$ which means that the q_0-term of the quaternion is 1 and the q_1- , q_2- and q_3- are all zero.

This product is also suitable for the product of a unit quaternion and an expanded vector.

A.2.3 Axis-angle rotation

The axis-angle representation of the rotation corresponds to a 3-vector, $\boldsymbol{\omega}$, where its length encodes the rotation angle, $\theta = ||\boldsymbol{\omega}||$, around the unitary axis represented by itself. As for the Euler angles case, the axis-angle representation does not allows direct computation of transformations. It can, however, be easily converted into a quaternion.

Axis-angle to Quaternion

Given an axis-angle rotation denoted $\boldsymbol{\omega} = \begin{bmatrix} \omega_0 & \omega_1 & \omega_2 \end{bmatrix}^T$ and its angle $\theta = ||\boldsymbol{\omega}||$, the quaternion $\mathbf{q}$ which represents $\boldsymbol{\omega}$ is computed as follows:

$$\mathbf{q} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ (\omega_0/\theta)\sin\left(\frac{\theta}{2}\right) \\ (\omega_1/\theta)\sin\left(\frac{\theta}{2}\right) \\ (\omega_2/\theta)\sin\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{A.10}$$

Quaternion to Axis-angle

Consider a unit quaternion $\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$. If $|q_0| = 1$, then the angle is $\theta = ||\boldsymbol{\omega}|| = 0$ and any unitary vector direction for the axis will do since there is no rotation. If $|q_0| < 1$, then $\boldsymbol{\omega}$ is given by

$$\boldsymbol{\omega} = (2\cos^{-1}(q_0)) \cdot \begin{bmatrix} q_1\sqrt{1-q_0^2} \\ q_2\sqrt{1-q_0^2} \\ q_3\sqrt{1-q_0^2} \end{bmatrix} \tag{A.11}$$

Appendix B

Tensor notation and Tensorial algebra

B.1 Short description of tensors

The necessity to deal with higher order bases than matrices and vectors entails the use of the standard tensor notation, also called Einstein notation. By the means of this notation, the representation of column and row vectors is done through covariant (e.g. x_i) and contravariant (e.g. x^j) indexes. A matrix $\mathbf{A}$, can then be represented as $\mathcal{A}^i_j$ where rows are noted by the contravariant index i and columns by the covariant j.

Consider now, a linear algebra operation $\mathbf{x}' = \mathbf{A}\mathbf{x}$. It can be alternatively represented in tensor notation as:

$$x'^i = \sum_j \mathcal{A}^i_j x^j = \mathcal{A}^i_j x^j \tag{B.1}$$

We define, in this convention, that indexes span over 1 to 3 and indexes repeated in the contravariant and covariant positions imply summation (in the example j). The number of indexes which defines the structure of a tensor is denoted as its valence.

The trifocal tensor is a third-order tensor composed of 27 elements arranged in a cube and represented by one covariant and two contravariants. It is possible to compute its three covariant slices through an algebraic expression from two 3×4 matrices denoted $\mathbf{A}$ and $\mathbf{B}$:

$$\mathbf{T}_i = \mathbf{b}_4 \mathbf{a}_i^T - \mathbf{a}_4 \mathbf{b}_i^T \tag{B.2}$$

where $\mathbf{a}_i$ and $\mathbf{b}_i$ are respectively the i^{th} columns of $\mathbf{A}$ and $\mathbf{B}$. $\mathbf{a}_4$ and $\mathbf{b}_4$ correspond to their fourth columns. Accordingly, this can be done by two tensor products and a subtraction:

$$\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k \tag{B.3}$$

This operation can be graphically represented as depicted in Fig. B.1. It shows, for example, that the first operand is composed of three slices in the k direction which are

the result of the multiplication of a_i^j by each element of b_4^k. The slices thus represent $a_i^j b_4^1$, $a_i^j b_4^2$ and $a_i^j b_4^3$. The subtraction is done on the operand slices, element by element in 3D.

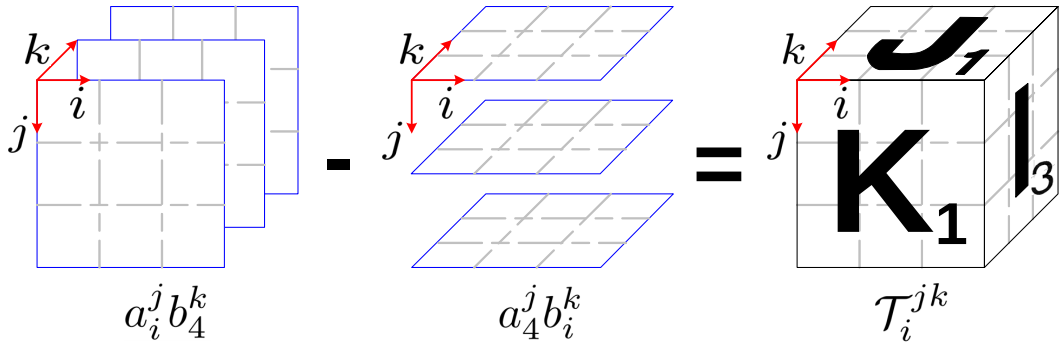


Figure B.1: *Graphical representation of the trifocal tensor computation*

B.1.1 Multiple view geometry application

In computer vision applications, we deal with tensors obtained from **A** and **B** which are computed from three canonical camera matrices. These tensors are used to achieve the transfer of geometrical entities between these different view points. In the sequel, two sample transfers are provided: Point transfer from first to third view via a plane in the second view and Point transfer from first to second view via a plane in the third view.

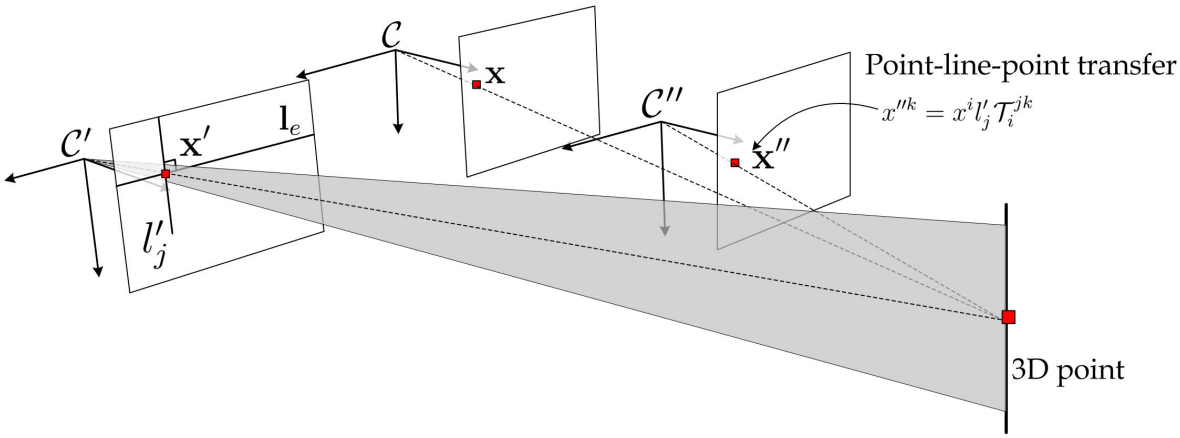


Figure B.2: *Point transfer from first to third view via a plane in the second view*

Point transfer from first to third view via a plane in the second view

Since the trifocal tensor representation encodes the geometrical constraints defined by three views, it is possible to obtain from it, a homography mapping (transfer) allowing the transfer between views of an observed point. Consider that a point $\mathbf{x}$ is observed in the first and second view ($\mathcal{C}$ and $\mathcal{C}'$) as shown in Fig. B.2. The image coordinates of

$\mathbf{x}$ in the third view, $\mathbf{x}''$ are given by:

$$x''^k = x^i l'_j \mathcal{T}_i^{jk} \tag{B.4}$$

where x^i is contravariant representation of the homogeneous image coordinates of $\mathbf{x}$, l'_j is the vertical image line passing through the corresponding point in the second view and x''^k is the transferred point coordinates.

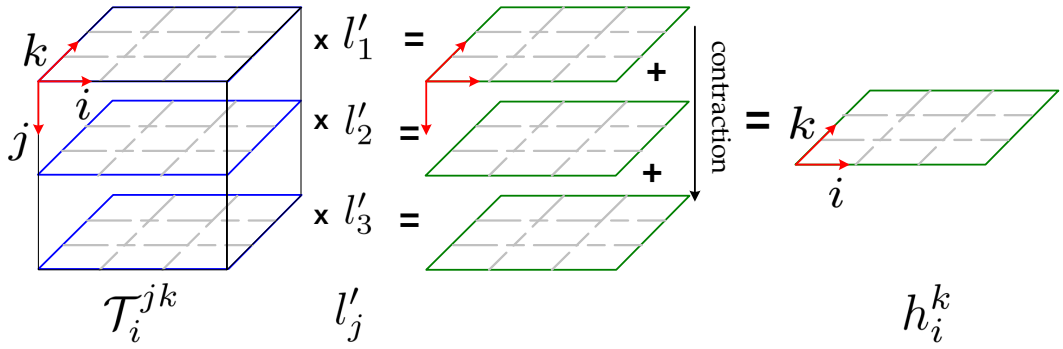
In Eq. B.4 the trifocal tensor, $\mathcal{T}_i^{jk}$ is initially "contracted" obtaining the homography, h_i^k, induced by the plane defined by the back-projection of l'_j,

$$h_i^k = l'_j \mathcal{T}_i^{jk} \tag{B.5}$$

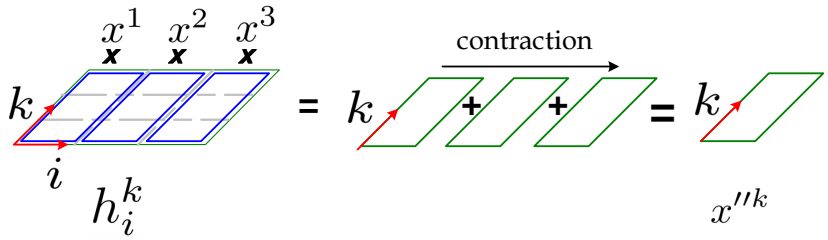
Then a new contraction is performed on the obtained homography resulting in the corresponding position of the observed point $\mathbf{x}$ transferred to the third view,

$$x''^k = h_i^k x^i \tag{B.6}$$

Fig. B.3 graphically illustrates the detailed operations.



(a) *First contraction given by Eq. B.5*



(b) *Second contraction given by Eq. B.6*

Figure B.3: *Trifocal tensor contraction*

Point transfer from first to second view via a plane in the third view

In a similar way to the former example, the contraction $l''_k \mathcal{T}_i^{jk}$ corresponds to the homography mapping between the first and the second view. This is induced by a

plane obtained through the back-projection of l''_k in the third view:

$$x'^j = h^j_i x^i \tag{B.7}$$

with $h^j_i = l''_k \mathcal{T}^{jk}_i$.

References

[AHB87] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[AK06] M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. *18th International Conference on Pattern Recognition, 2006. ICPR 2006.*, 3:1063 – 1068, 2006.

[BA03] James P. Bliss and Sarah A. Acton. Alarm mistrust in automobiles: how collision alarm reliability affects driving. *Applied Ergonomics*, 34(6):499 – 509, 2003.

[Bab94] Chris Baber. *Human factors in Alarm Design*, chapter Psychological aspects of conventional in-car warning devices, pages 193–205. Taylor & Francis, Inc., Bristol, PA, USA, 1994.

[BAHH92] James R. Bergen, P. Anandan, Keith J. Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. *European Conference on Computer Vision*, pages 237–252, 1992.

[BBS88] Henk A. Blom and Yaakov Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33:780 – 783, 1988.

[BCG+09] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H.G. Jung. A new approach to urban pedestrian detection for automatic braking. *Journal of Intelligent Vehicles Systems*, 10(4):594–605, 2009.

[BETG08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding (CVIU)*, 110:346–359, 2008.

[BH87] Steven D. Blostein and Thomas S. Huang. Error analysis in stereo determination of 3d point positions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:752–765, 1987.

[BK08] Gary Bradski and Adrian Kaehler. *Learning OpenCV Computer Vision with OpenCV*. O'Reilly, 2008.

[Bla86] Samuel S. Blackman. *Multiple Target Tracking with Radar Applications*. Dedham, MA, Artech House, 1986.

[BM92] Paul J. Besl and Neil D. Mckay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239, 1992.

[BM04] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56:221–255, 2004.

[Bou02] Jean-Yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. Technical report, Intel Corporation Microprocessor Research Labs, 2002.

[BP99] Samuel S. Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Incorporated, 1999.

[BSFC08] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. *ECCV*, 2008.

[BT74] A. E. Beaton and J.W. Tukey. *The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data*. Technometrics, 1974.

[CCG06] Nico Cornelis, Kurt Cornelis, and Luc Van Gool. Fast compact city modeling for navigation pre-visualization. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1339–1344, 2006.

[CMR07] Andrew Comport, Ezio Malis, and Patrick Rives. Accurate quadrifocal tracking for robust 3d visual odometry. *IEEE International Conference on Robotics and Automation*, pages 40–45, April 2007.

[CMR10] Andrew Comport, Ezio Malis, and Patrick Rives. Real-time quadrifocal visual odometry. *International Journal of Robotics Research*, 29:245–266, 2010.

[CNPC08] Cindy Cappelle, Maan E. El Najjar, Denis Pomorski, and Francois Charpillet. Multi-sensors data fusion using dynamic bayesian network for robotised vehicle geo-localisation. *IEEE Fusion*, 2008.

[CS09] Boguslaw Cyganek and J. Paul Siebert. *An Introduction to 3D Computer Vision Techniques and Algorithms*. John Wiley and Sons, 2009.

[Dic07] Ernst D. Dickmanns. *Dynamic Vision for Perception and Control of Motion*. Springer, 2007.

[DJ08] Gregory Dudek and Michael Jenkin. *Springer Handbook of Robotics*, chapter Inertial Sensors, GPS, and Odometry, pages 477–490. Springer Berlin Heidelberg, 2008.

[DKF05] Romain Dupont, Renaud Keriven, and Philippe Fuchs. An improved calibration technique for coupled single-row telemeter and ccd camera. *The International Conference on 3-D Digital Imaging and Modeling*, 2005.

[DSS01] Klaus C. J. Dietmayer, Jan Sparbert, and Daniel Streller. Model based object classification and object tracking in traffic scenes from range images. *IEEE Intelligent Vehicles Symposium*, 1:1–6, 2001.

[DWB06] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localisation and mapping (slam). *IEEE Robotics & Automation Magazine*, 13:99–110/108 – 117, 2006.

[Faw06] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[Fay09] Fadi Fayad. *Gestion de la confiance dans un systeme de fusion multi-sensorielle. Application a la detection de pietons en situations routieres*. PhD thesis, Universite de Technologie de Compiegne, 2009.

[FC02] Vincent Fremont and Ryad Chellali. Direct camera calibration using two concentric circles from a single view. *International Conference on Artificial Reality and Telexistence*, 2002.

[FC07] Fadi Fayad and Veronique Cherfaoui. Tracking objects using a laser scanner in driving situation based on modeling target shape. *IEEE Intelligent Vehicles Symposium*, 1:44–49, 2007.

[FTV99] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications Manuscript*, 1:1–8, 1999.

[GGS94] Walter Gander, Gene H. Golub, and Rolf Strebel. Least-squares fitting of circles and ellipses. 1994.

[GM06] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision*, 73(1):41–59, Juin 2006.

[GT05] Tarak Gandhi and Mohan Trivedi. Parametric ego-motion estimation for vehicle surround analysis using an omnidirectional camera. *Machine Vision and Applications*, 16:85–95, 2005.

[GvL96] Gene H. Golub and Charles F. van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, USA, 3 edition, 1996.

[HGJD09] C. Hughes, M. Glavin, E. Jones, and P. Denny. Wide-angle camera technology for automotive applications - a review. *Intelligent Transportation Systems*, 3(1):19–31, 2009.

[HS81] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. *Proceedings fo The Fourth Alvey Vision Conference*, 1:147–151, 1988.

[Hub81] P.J. Huber. *Robust Statistics*. John Wiley, 1981.

[HZ03] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision. Second Edition*. Cambridge, 2003.

[Jul02] Simon J. Julier. The scaled unscented transformation. *American Control Conference*, 6:4555–4559, 2002.

[Kal60] Rudolf Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, (82):35–45, 1960.

[KD03] Nico Kaempchen and Klaus Dietmayer. Data synchronization strategies for multi-sensor fusion. *Proceedings of the 10th World Congress on Intelligent Transport Systems and Services*, (T2250), 2003.

[KGK05] Jun-Sik Kim, Pierre Gurdjos, and In-So Kweon. Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4), 2005.

[KLD09] Anupama Krishnan, Chris Lewis, and Day Dwight. Vision system for identifying road signs using triangulation and bundle adjustment. *IEEE International Conference on Intelligent Transportation Systems*, 1:36–41, 2009.

[KO04] Kenichi Kanatani and Naoya Ohta. Automatic detection of circular objects by ellipse growing. *Int. J. Image Graphics*, 4(1):35–50, 2004.

[KZD04] Nico Kaempchen, Markus Zocholl, and Klaus C.J. Dietmayer. Spatio-temporal segmentation using laserscanner and video sequences. *Proceedings Pattern Recognition 26th DAGM Symposium*, 3175:367–374, 2004.

[LCCG07] Bastian Leibe, Nico Cronelis, Kurt Cornelis, and Luc Van Gool. Dynamic 3d scene analysis from a moving vehicle. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, 1, 2007.

[Lev44] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.

[LH06] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. *International Conference on Pattern Recognition*, pages 630–633, 2006.

[LHL08] Martin E. Liggins, David L. Hall, and James Llinas. *Handbook of Multi-Sensor Data Fusion*. CRC Press, 2008.

[LK81] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of the 1981 DARPA Imaging Understanding Workshop*, 1:121–130, 1981.

[Lou] Manolis Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in c/c++.

[LOV] ANR-Predit project Logiciel d'Observation des Vulnerables LOVe. http://love.univ-bpclermont.fr.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 20:91–110, 2004.

[LRGA05] Raphael Labayrade, Cyril Royere, Dominique Gruyer, and Didier Aubert. Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner. *Autonomous Robots*, 19:117–140, 2005.

[Lup93] Robert Lupton. *Statistics in Theory and Practice*. Princeton University Press, 1993.

[Mac67] J. MacQueen. Some methods for classification and analisys multivariate observations. *Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–297, 1967.

[Mar63] D.W. Marquardt. An algorithm for the least-squares estimation of non-linear parameters. *SIAM Journal of Applied Mathematics*, 11:431–441, 1963.

[Mei07] Christopher Mei. *Couplage Vision Omnidirectionnelle et Telemetrie Laser pour la Navigation en Robotique / Laser-Augmented Omnidirectional Vision for 3D Localisation and Mapping.* PhD thesis, INRIA Sophia Antipolis, Project-team ARobAS, 2007.

[MON09] T. Miyasaka, Y. Ohama, and Y. Ninomiya. Ego-motion estimation and moving object tracking using multi-layer lidar. *IEEE Intelligent Vehicles Symposium*, 1:151–156, 2009.

[Nis04] David Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:756–777, 2004.

[NKL08] Fawzi Nashashibi, Ayoub Khammari, and Claude Laurgeau. Vehicle recognition and tracking using a generic multisensor and multialgorithm fusion approach. *International Journal of Vehicle Autonomous Systems*, 6:134–154, 2008.

[NMTS07] Viet Nguyen, Stefan Gachter Agostino Martinelli, Nicola Tomatis, and Roland Siegwart. A comparison of line extraction algorithms using 2d range data for indoor mobile robotics. *Autonomous Robots*, 23(2):97–111, 2007.

[NNB04] David Nister, Oleg Naroditsky, and James Bergen. Visual odometry. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:652–659, 2004.

[NNB06] David Nister, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1), 2006.

[Ope] Intel Corporation Open Computer Vision Library OpenCV. http://sourceforge.net/projects/opencvlibrary.

[PLR+06] Mathias Perrollaz, Raphael Labayarde, Cyril Royere, Nicolas Hautiere, and Didier Aubert. Long range obstacle detection using laser scanner and stereo vision. *IEEE Intelligent Vehicles Symposium*, 1:182–187, 2006.

[PT09] Anna Petrovskaya and Sebastian Thrun. Model based vehicle tracking in urban environments. *IEEE International Conference on Robotics and Automation, Workshop on Safe Navigation*, 1:1–8, 2009.

[SAF] SAFESPOT. Cooperative vehicles and road infraestructure for road safety. http://www.safespot-eu.org/.

[SB00] Yaakov Bar Shalom and William Dale Blair. *Multitarget/Multisensor Tracking: Applications and Advances*. Artech House Publishers, 2000.

[SF06] Roland Schulz and Kay Furstenberg. *Adavanced Microsystems for Automotive Applications*, chapter Laserscanner for Multiple Applications in Passenger Cars and Trucks, pages 129–141. Springer Berlin Heidelberg, 2006.

[SFS09] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. *IEEE International Conference on Robotics and Automation ICRA*, 1:1, 2009.

[Sha98] Craig M. Shakarji. Least-squares fitting algorithms of the nist algorithm testing system. Technical Report 6, National Institute of Standards and Technology, 1998.

[SHRE00] C. Stiller, J. Hipp, C. Rossig, and A. Ewald. Multisensor obstacle detection and tracking. *Image and Vision Computing*, 18(5):389–396, 2000.

[SHS07] D. Scaramuzza, A. Harati, and R. Siegwart. Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[SKG09] Bruno Siciliano, Oussama Khatib, and Frans Groen. *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*. Springer Berlin / Heidelberg, 2009.

[Sol07] Joan Sola. *Towards Visual Localization, Mapping and Moving Objects Tracking by a Mobile Robot: A Geometric and Probabilistic Approach*. PhD thesis, Institut National Politechnique de Toulouse, 2007.

[SS01] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2001.

[ST94] Jianbo Shi and Carlo Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593 – 600, 1994.

[Ste99] Charles V. Stewart. Robust parameter estimation in computer vision. *Society for Industrial and Applied Mathematics*, 41(3):513–537, 1999.

[Sti01] Christoph Stiller. *Intelligent Vehicle Technologies, Theory and Applications*, chapter Towards Intelligent Automotive Vision Systems, pages 113–128. Butterworth Heinemann, 2001.

[SW00] Amnon Shashua and Lior Wolf. On the structure and properties of the quadrifocal tensor. In *ECCV (1)*, pages 710–724, 2000.

[SW09] Ciaran Simms and Denis Wood. *Pedestrian and Cyclist Impact*, volume 166, chapter Pedestrian and Cyclist Injuries, pages 5–30. Springer, 2009.

[TCD+06] C. Tessier, C. Cariou, C. Debain, F. Chausse, R. Chapuis, and C. Rousset. A real-time, multi-sensor architecture for fusion of delayed observations: Application to vehicle localization. *IEEE Intelligent Transportation Systems Conference*, pages 1316–1321, 2006.

[TFB00] Sebastian Thrun, Dieter Fox, and Wolfram Burgard. *Probabilistic Robotics*. MIT Press, 2000.

[Tri97] Bill Triggs. Autocalibration and the absolute quadric. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, page 609, Washington, USA, 1997. IEEE Computer Society.

[vdMFDG02] Wannes van der Mark, DaniÂšel Fontijne, Leo Dorst, and Frans C.A. Groen. Vehicle ego-motion estimation with geometric algebra. *Proceedings IEEE Intelligent Vehicle Symposium, Versialles*, 1:18–20, 2002.

[VJS03] Paul Viola, Michael J. Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *IEEE International Conference on Computer Vision*, 1:734–741, 2003.

[WTH+07] Chieh-Chih Wang, Charles Thorpe, Martial Herbert, Sebastian Thrun, and Hugh Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *International Journal of Robotics Research*, 26:889–916, 2007.

[XJ02] Yonghong Xie and Qiang Ji. A new efficient ellipse detection method. *International Conference on Pattern Recognition*, 2:957–960, 2002.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE T. Pattern. Anal.*, 22(11):1330–1334, 2000.

[ZP04] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). *Proceedings. 2004 IEEE/RSJ. Intelligent Robots and Systems, 2004.*, 2004.

[ZXJ+09] Huijing Zhao, Long Xiong, Zhigang Jiao, Jinshi Cui, and Hongbin Zha. Sensor alignment towards an omni-directional measurement using an intelligent vehicle. *IEEE Intelligent Vehicles Symposium*, 1:292–298, 2009.